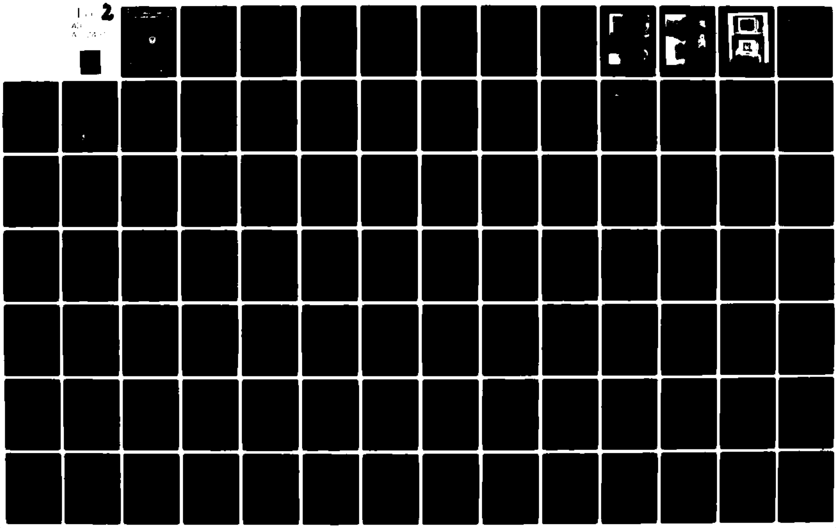
AD-A107 461   ARMY PROJECT MANAGER FOR TRAINING DEVICES ORLANDO FL   F/G 19/5
SIMULATED TANK ANTI-ARMOR GUNNERY SYSTEM (STAGS).(U)
NOV 81 A MARSHALL, B SHAW, H TOWLE

UNCLASSIFIED   PM TRADE-RE-0010   NL

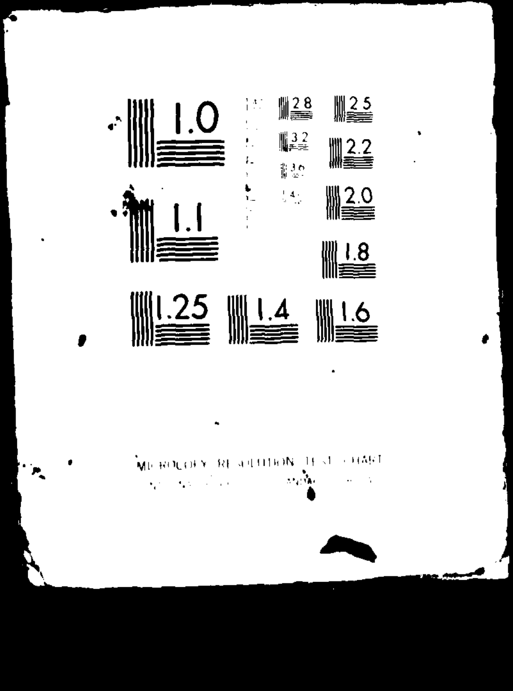
1 of 2
AD
A107461



SSIFIED

1 OF 2

AD-
A107481



MICROCOPY RESOLUTION TEST CHART

2

SIMULATED TANK ANTI-ARMOR GUNNERY SYSTEM (STAGS-D)

BY Albert Marshall
Dr. Herbert Towle
Bon Shaw
Gary Bond
George Siragusa

U.S. ARMY MATERIEL DEVELOPMENT AND READINESS COMMAND

DRAGON TRAINER

PREPARED FOR

U.S. ARMY PROJECT MANAGER FOR TRAINING DEVICES
NAVAL TRAINING EQUIPMENT CENTER (NTEC)
ORLANDO, FLORIDA 32813

BY

ADVANCED SIMULATION CONCEPTS LABORATORY
SIMULATION TECHNOLOGY BRANCH

NOVEMBER 1981

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
PM-TRADE-RE-0010

2. GOVT ACCESSION NO.
AD-A107481

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Simulated Tank Anti-Armor Gunnery System (STAGS),

5. TYPE OF REPORT & PERIOD COVERED
Interim Report.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Albert Marshall, Bon Shaw, ~~Dr.~~ Herbert Towle
George Siragusa ~~and~~ Gary Bond

8. CONTRACT OR GRANT NUMBER(s)
Project 1785

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Training Equipment Center
N-73
Orlando, FL 32813

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
62727A230

11. CONTROLLING OFFICE NAME AND ADDRESS (305/646-5771)
U. S. Army (DRCPM-TND-RE)
Project Manager for Training Devices, Orlando, FL 32813

12. REPORT DATE
October 1981

13. NUMBER OF PAGES

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Trainer
Infantry
Anti-Armor
Anti-Tank
DRAGON
Missile
Microprocessor
TOW
Multiprocessor

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a missile flight simulator research model developed to train DRAGON gunners. It is also being adapted to a variety of similar anti-armor weapons. The system employs a terrain board with enemy armored vehicles moving in a variety of attack scenarios. When the gunner fires the missile, hears computer generated rocket sounds and experiences the weight loss, recoil and smoke of the missile launch. When the smoke clears, he views the missile in the sight as well as the target. The gunner's aiming error is measured

20. using a micro-processor-controlled diode matrix array. The matrix detector senses an IR emitting diode which is located on the miniature target. The flight equations of motion for the missile are solved by a 16 bit microprocessor every 0.02 seconds in each axis using gunner aiming error, target position, gravity, drag and side thruster accelerations as inputs. A second coordinated 16 bit processor controls a display that plots both vertical and horizontal aiming error for analysis of the gunner's performance by an instructor. Experienced DRAGON gunners have tested the system and attested to the realism and training potential.

SUMMARY

The Simulated Tank and Anti-Armor Gunnery Systems (STAGS) is an electro-optic based, multiprocessor controlled, training device that enables training of DRAGON gunners at a reasonable cost. In a short period of time a gunner can be subjected to a variety of target scenarios and his performance can be analyzed in real-time by an instructor. When the trainee fires the training device he hears the initial explosion of the rocket motor. He experiences a weight loss due to the simulated rocket exiting the tube as well as a recoil force. Momentarily his view in the sight is obscured by simulated smoke. The trainee must overcome such launch transients and smoothly track the target and ignore the simulated missile which he can see in his sight. Thruster rocket firing sounds are included as well as the final hit or ground impact explosions. A visual indication of missile impact is also inserted into the gunners sight.

During missile flight the instructor can monitor two displays. These displays show:

(1) The gunner's sight picture, the DRAGON missile's location, and

(2) a plot of gunner aiming error versus time and the gunner error tracking limit envelopes. Thruster firings are annotated on the display.

This system uses a 16 bit microprocessor to solve the flight equations every 0.02 seconds in each axis using the gunner's aiming error, target position, gravity, drag and thruster rocket acceleration as inputs. The solution also incorporates the dynamic performance of the tracker.

A prototype model was constructed by the Advanced Simulations Concepts Laboratory, Naval Training Equipment Center, Orlando, Florida for the U.S. Army Project Manager for Training Devices (PM TRADE) and the U.S. Marine Corps.

This model has been successfully evaluated by experienced DRAGON gunner teams from both the U.S. Army and U.S. Marine Corps.

Demonstrations and evaluations are planned during FY 82 using this device. Expansion of the device to include TOW is also planned.

The PM TRADE Project Manager is A. Boudreaux. The authors wish to thank him for the helpful assistance he gave during this program.

TABLE OF CONTENTS

PAGE NUMBER

Appendix

LIST OF ILLUSTRATIONS PAGE NUMBER

SECTION I

INTRODUCTION

Training in the firing of modern anti-armor weapons is expensive. Each live round costs thousands of dollars.

This report describes a system that uses advanced electro-optics and microprocessor technology to enable training of DRAGON gunners at a reasonable cost.

The DRAGON is a command-to-line-of-sight guided missile system. Fired from a recoilless launcher, the missile is tracked optically and guided automatically to the target by electrical impulses transmitted via a wire link. Firing the DRAGON missile is accomplished by depressing the safety and squeezing the trigger. No other action is required of the gunner except to keep the sight cross hairs on the target. However, to score a hit the trainee must overcome many perturbations that can spoil his track.

When the trainee fires the training device he hears the initial explosion of the rocket motor. He experiences a simulated weight loss due to the rocket exiting the tube as well as a recoil force. Momentarily he is blinded in the sight by simulated smoke. The trainee must overcome such launch transients. He must smoothly track the target and ignore the simulated missile which he can see in his sight. Thruster rocket firing sounds are included as well as the final hit or ground impact explosions. A visual indication of hit is also inserted into the gunne.'s sight.

During missile flight the instructor can monitor two displays. These displays show:

(1) The gunner's sight picture and the DRAGON's location.

(2) Plots of gunner aiming error in azimuth and elevation versus time and the gunner error tracking limit envelopes. Thruster firings are annotated on the display and the number of thruster firings, actual versus ideal are recorded.

(3) The instructor can recall four additional plots after the mission is over: gunner aiming error and missile location in azimuth versus time; gunner aiming error and missile location in elevation versus time.

This system uses a 16 bit microprocessor to solve the flight equations every 0.02 seconds in each axis using the gunner's aiming error, target position, gravity, drag and thruster rocket acceleration as inputs. The solution also incorporates the dynamic performance of the DRAGON tracker.

Key features of the system are summarized below.

- Target hit or miss determined by solving DRAGON flight equations in real time
- Smoke obscuration

- Recoil
- Weight Loss
- Missile superimposed on gunner's view of scenario
- Sounds - thruster firing, launch, hit and miss explosions, gyro wind-up
- Gunner aiming errors versus time displayed in real time
- Missile position versus time which can be recalled along with gunner aiming errors in azimuth and elevation for analysis
- Expensive tank target and special range is not required for training
- Number of thruster rocket firing's ideal versus actual displayed
- Portable
- Record and play back capability
- Can operate with and without an instructor
- Can be used to simulate night firings with a thermal sight
- Gunner's pull down force on DRAGON launcher, and eye piece pressure is indicated
- Variety of target speeds and motions simulated
- Trainer flies like real missile because of computation of flight parameters

Photographs of the system are shown in Figure I-1 through I-4 and illustrate the student station, instructors console, and the terrain board.

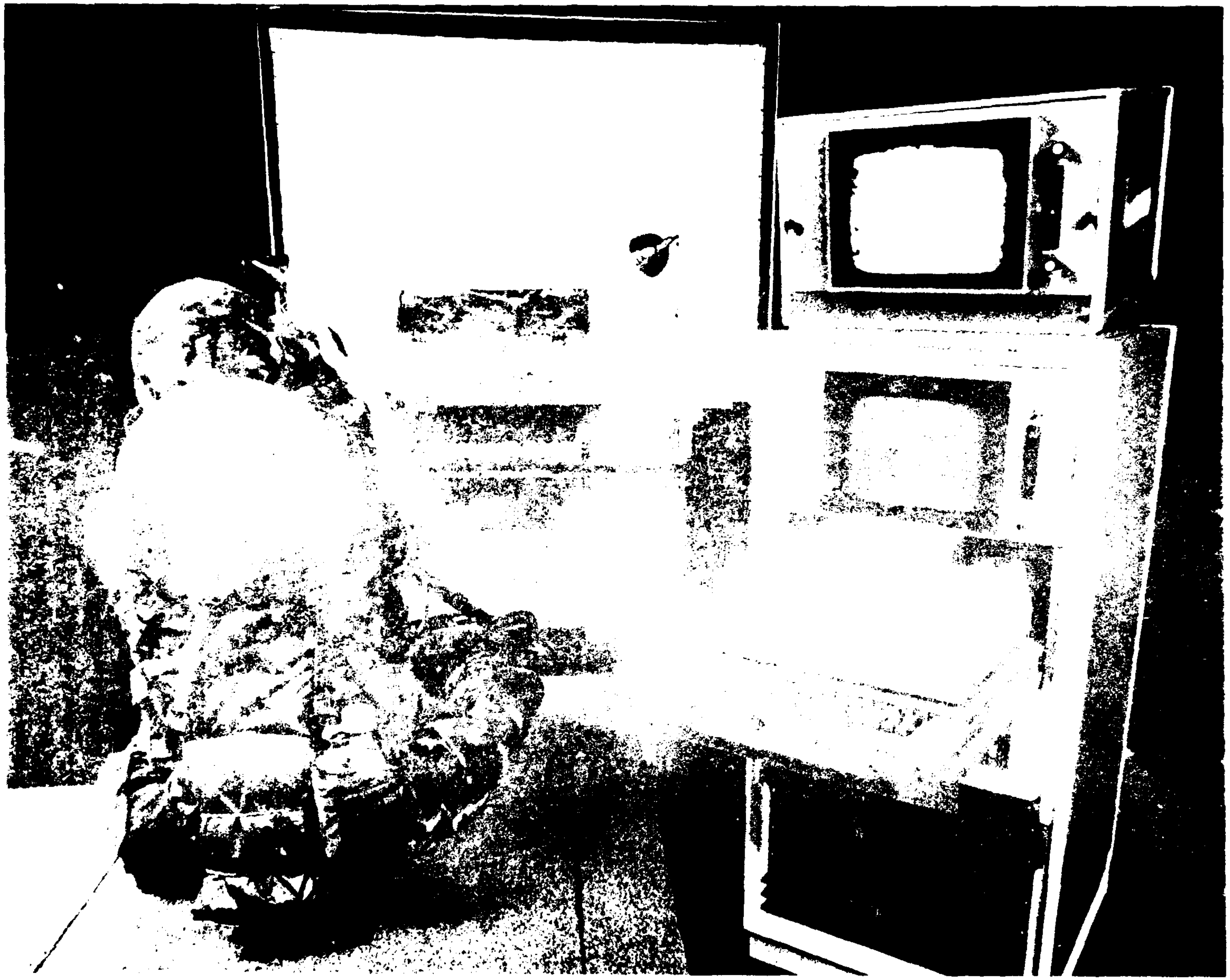


FIGURE I-1 DRAGON SYSTEM

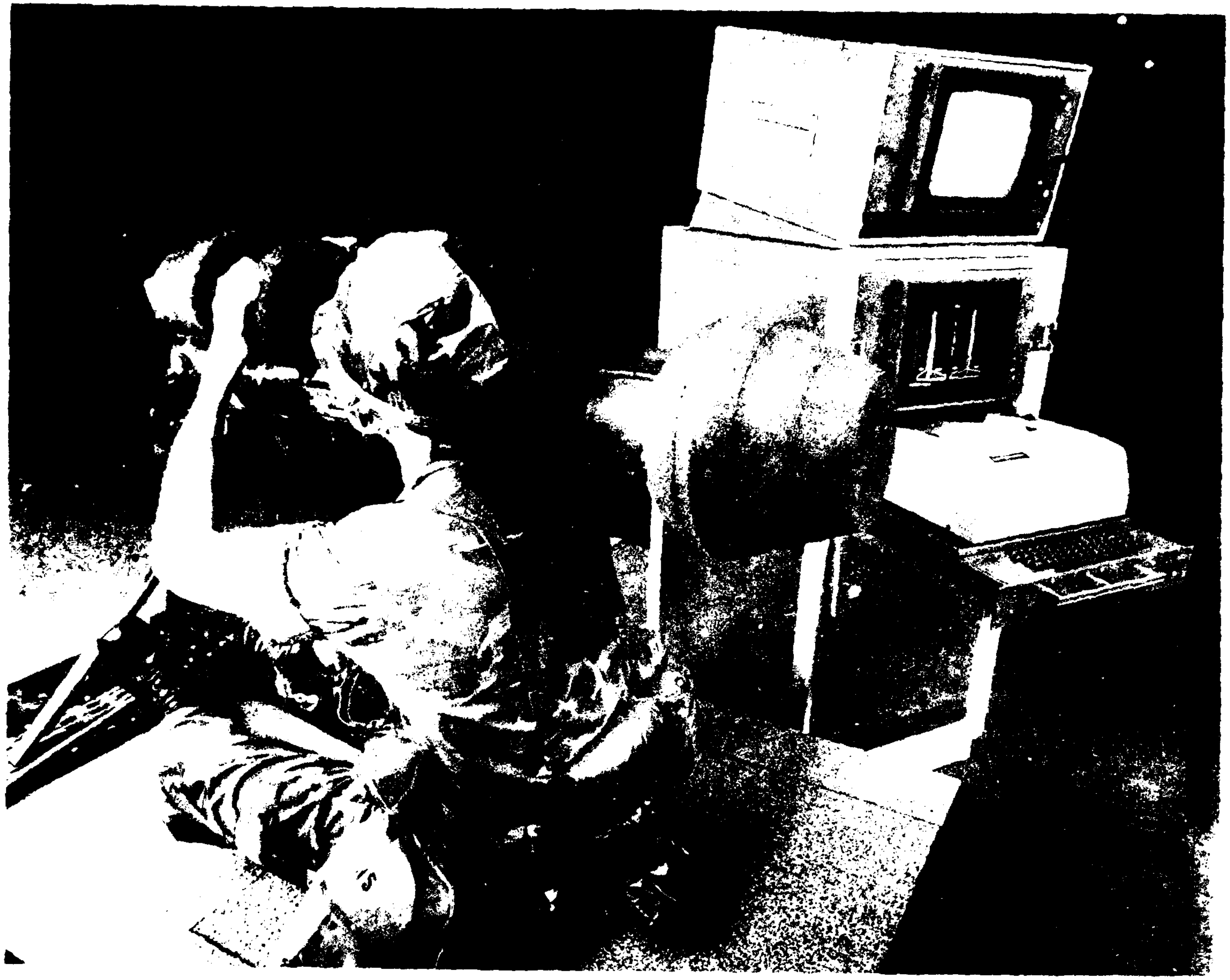


FIGURE II-2. STUDENT STATION AND INSTRUCTOR'S CONSOLE

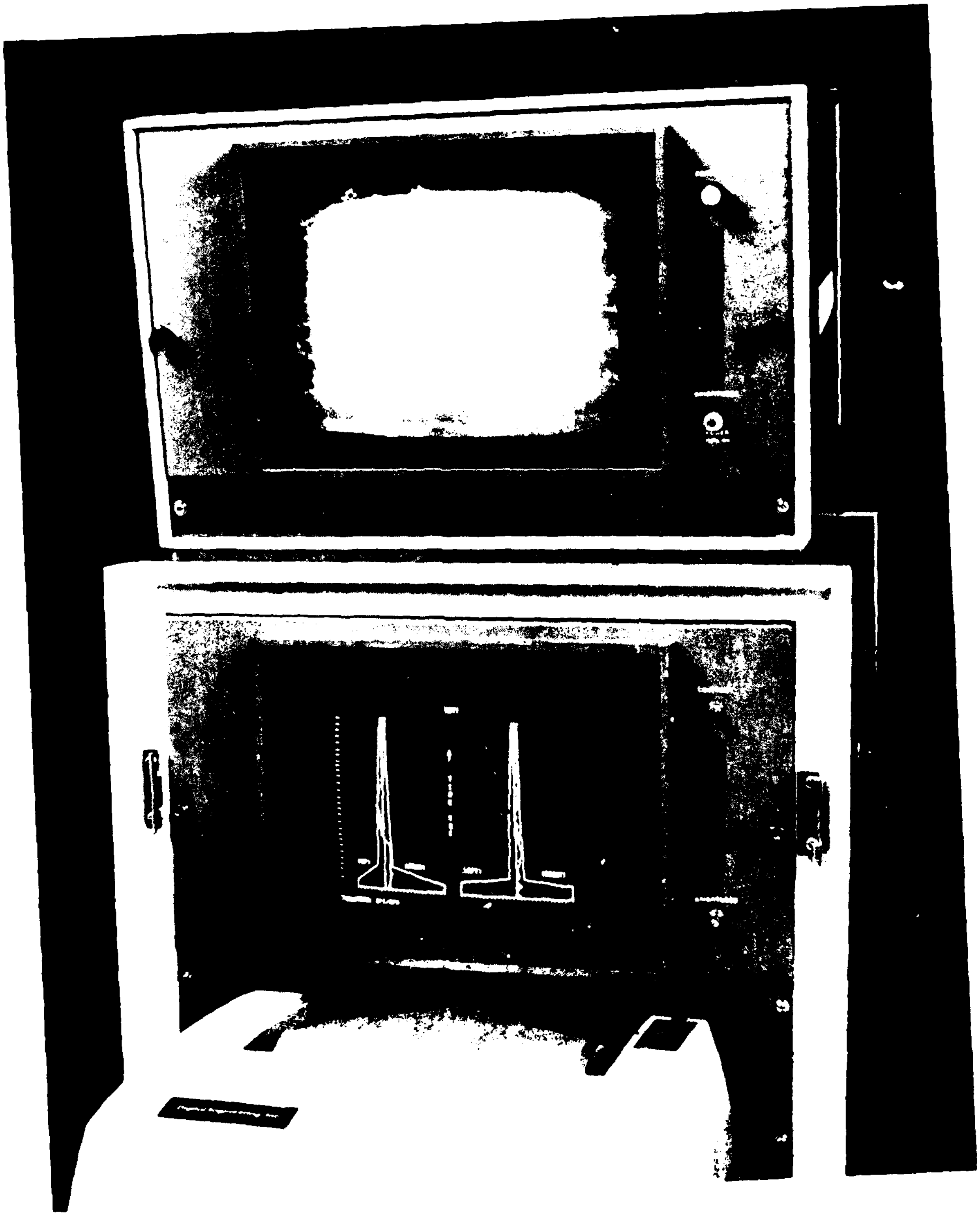


FIGURE I-3 INSTRUCTION CONSOLE

SECTION II

SYSTEM DESCRIPTION

The system block diagram is shown in Figure II-1.

Targets in this system are 1/120 minature models. Model targets were chosen because they have better resolution than either computer generated imagery or a movie display. DRAGON utilizes a 6X sighting scope. In other weapons even higher power sighting scopes are utilized, thus demanding a high resolution visual scenario.

Models are moved on a terrain board using a stepper motor under the control of a single chip microprocessor. The engagement scenario is stored in the Personnel Interface Processor (PIP) and is selected on the instructor's console by an input terminal. The stored scenario program contains the tank target's velocities, directions and range. Scenario data are provided to the DRAGON Flight Simulator Processor (DFS). At the center of mass of the scaled target is an Infrared Emitting Diode (IRED). Located in the DRAGON launch tube is a photo diode array camera to sense the IRED. The IRED is invisible to the human eye. This 100 x 100 matrix camera is boresighted to the gunner's sight telescope and used to determine the gunner's aiming error (GAE) which is input to the DRAGON Flight Simulator processor. This processor solves the DRAGON flight equations and provides DRAGON status to the Personnel Interface Processor (PIP). The PIP controls the graphics units which inserts the missile, smoke, explosion, etc., into the gunner's sight. This processor also controls the Gunner Aiming Error (GAE) display on the Instructor's Console. This display plots GAE versus time, in real time. The DRAGON Flight Simulator Processor produces launch and target explosions, thruster rocket firings and gyro noises. The thruster rocket firings are delayed to allow for the speed of sound versus the visual phenomena of the rocket firing which is optically inserted in the DRAGON gunner's sight. Rocket thruster noises are attenuated as a function of distance.

A closed circuit TV (CCTV) is located on the DRAGON tube and boresighted to the gunner's 6X sight. The Gunner's Sight Picture Display is located on the instructor's console. The DRAGON rocket as seen by the trainee is also mixed into the gunner's sight picture visual display.

An indicator on the instructor's console indicates the ammount of pulldown pressure on the launcher tube. Automatic computer controlled boresight is also incorporated.

Data print-outs for both a hit and miss training session are shown in Figures II-2 and II-3. Note that the rocket thruster firings are annotated on the left hand margin with dashes indicating the firing of a thruster rocket pair. Time between thruster firings are a function of the rate of change of gunner aiming error.

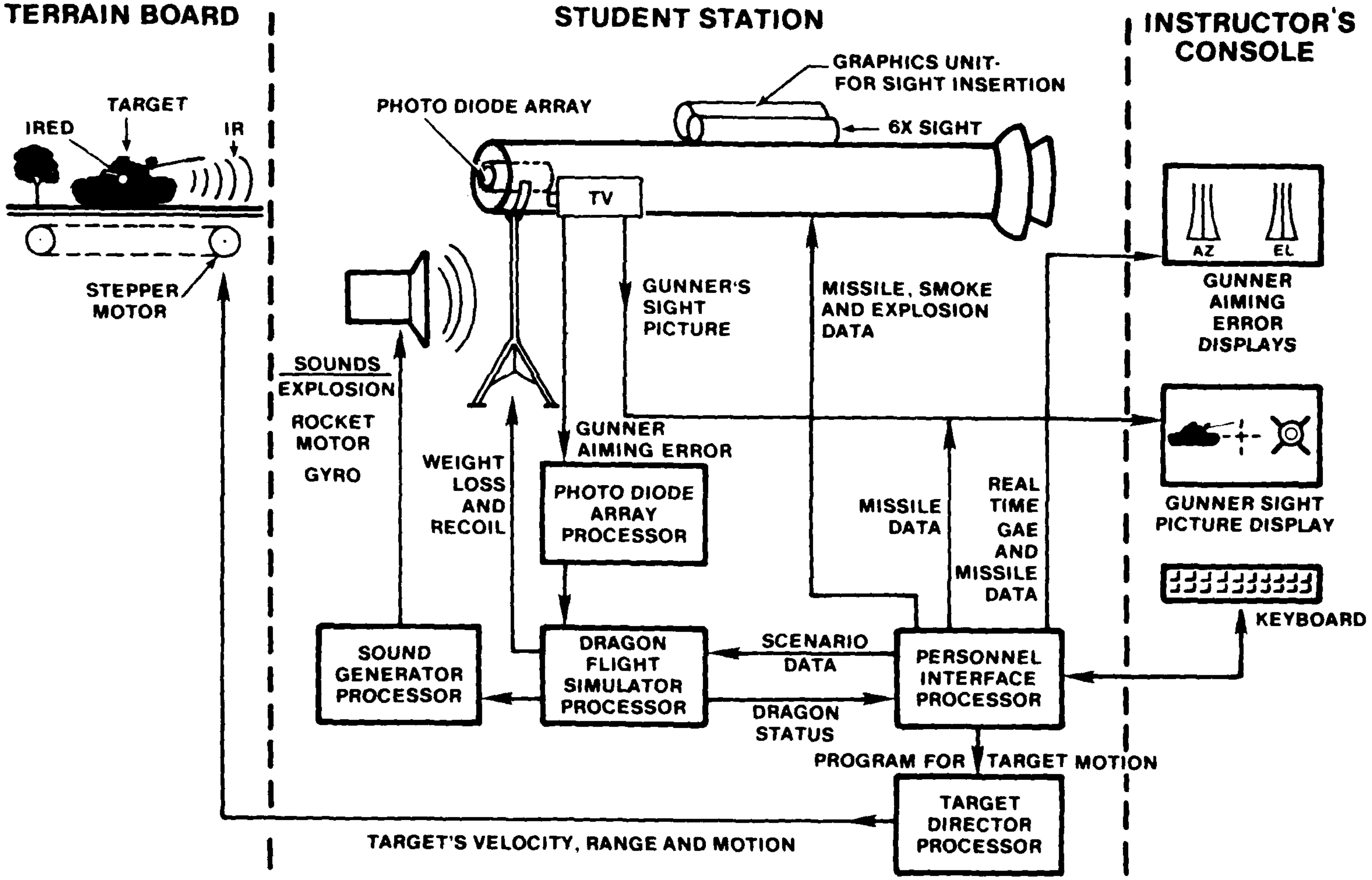

TERRAIN BOARD
STUDENT STATION
INSTRUCTOR'S CONSOLE
TARGET
IRED
IR
STEPPER MOTOR
PHOTO DIODE ARRAY
GRAPHICS UNIT- FOR SIGHT INSERTION
6X SIGHT
TV
GUNNER'S SIGHT PICTURE
MISSILE, SMOKE AND EXPLOSION DATA
AZ
EL
GUNNER AIMING ERROR DISPLAYS
SOUNDS
EXPLOSION
ROCKET MOTOR
GYRO
WEIGHT LOSS AND RECOIL
GUNNER AIMING ERROR
PHOTO DIODE ARRAY PROCESSOR
MISSILE DATA
REAL TIME GAE AND MISSILE DATA
GUNNER SIGHT PICTURE DISPLAY
KEYBOARD
SOUND GENERATOR PROCESSOR
DRAGON FLIGHT SIMULATOR PROCESSOR
SCENARIO DATA
DRAGON STATUS
PERSONNEL INTERFACE PROCESSOR
PROGRAM FOR TARGET MOTION
TARGET DIRECTOR PROCESSOR
TARGET'S VELOCITY, RANGE AND MOTION

FIGURE II-1 SYSTEM BLOCK DIAGRAM

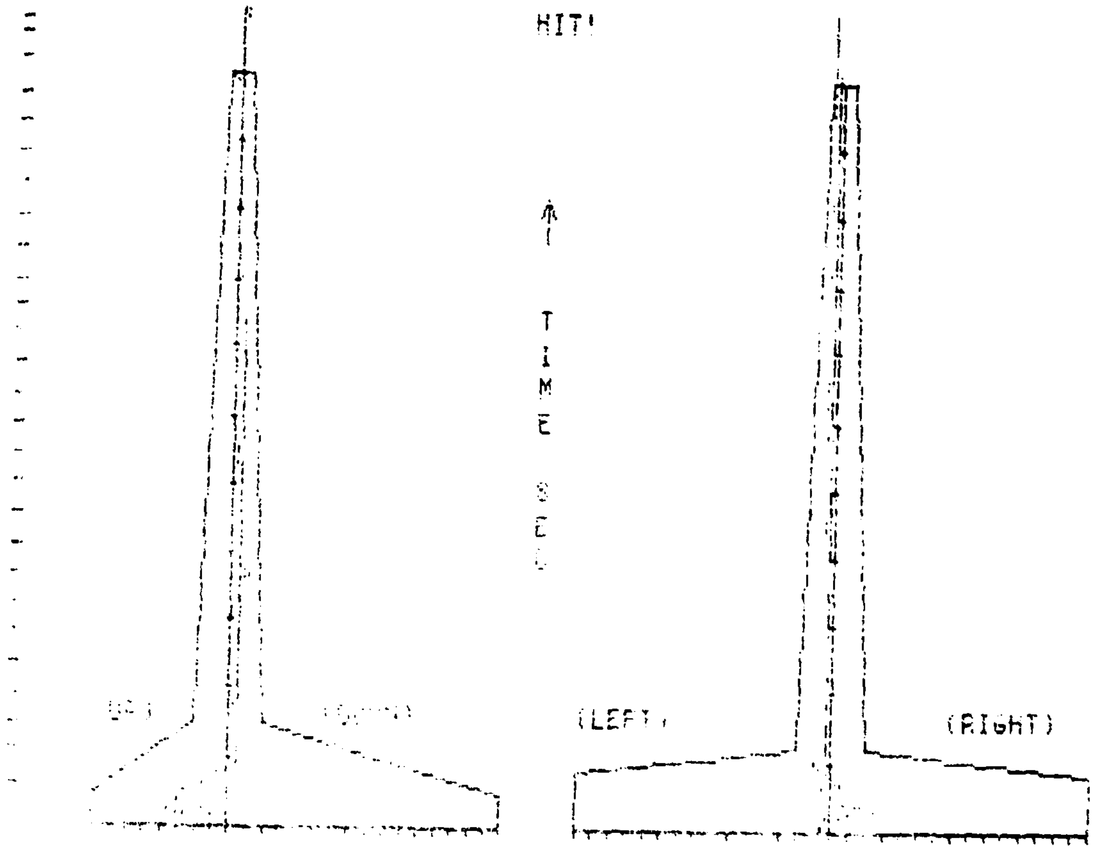


FIGURE II-2 (1 of 3) GAE vs Time
(Vertical & horizontal Planes)
- HIT

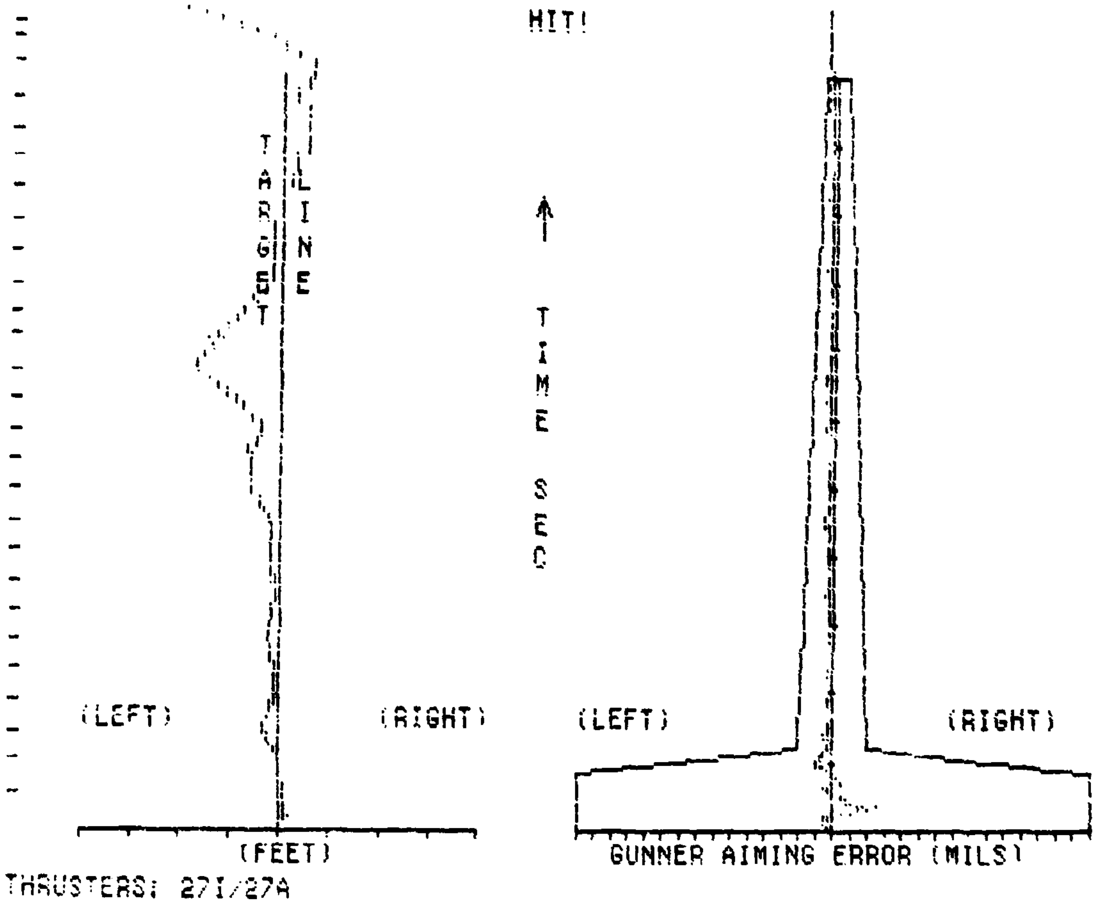


FIGURE II-2 (2 of 3) Missile Location vs Time (Horizontal Plane) - HIT

GAE vs Time (Horizontal Plane)

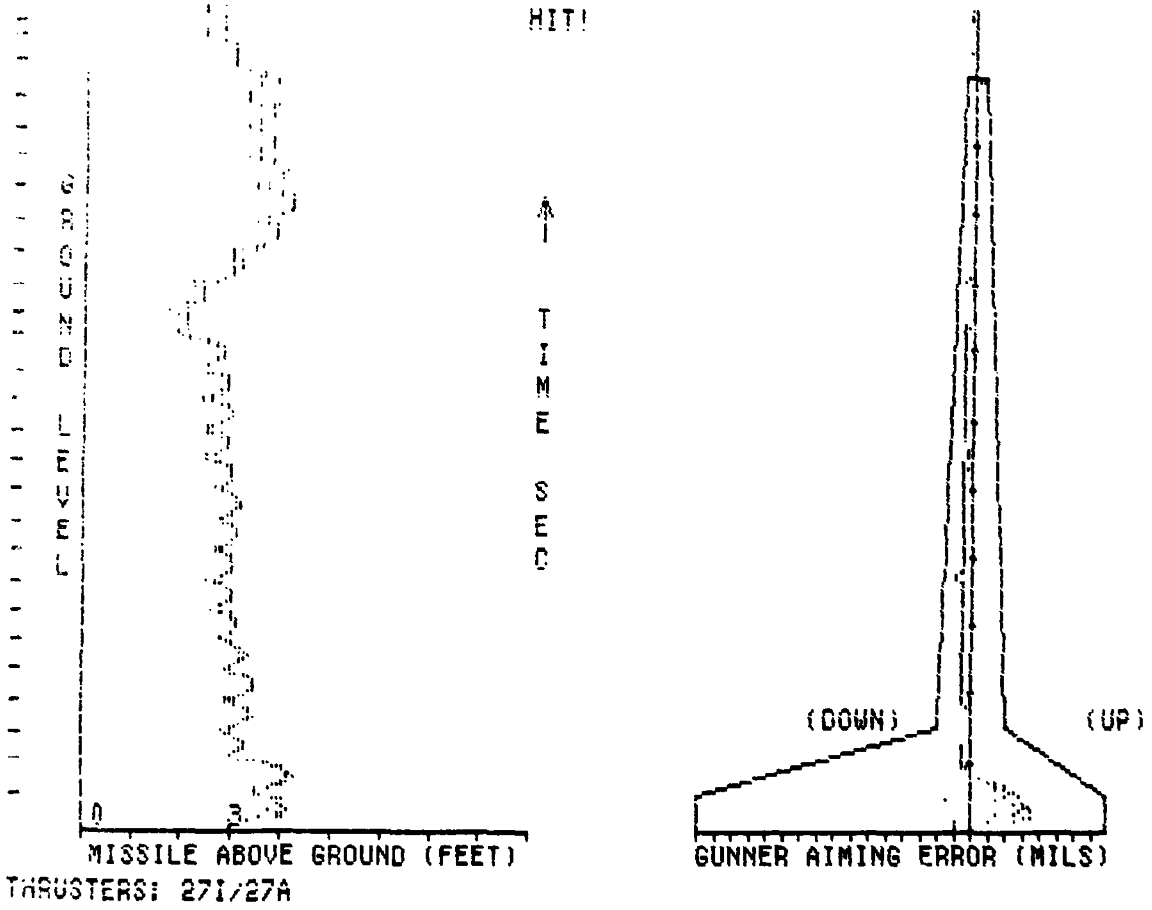
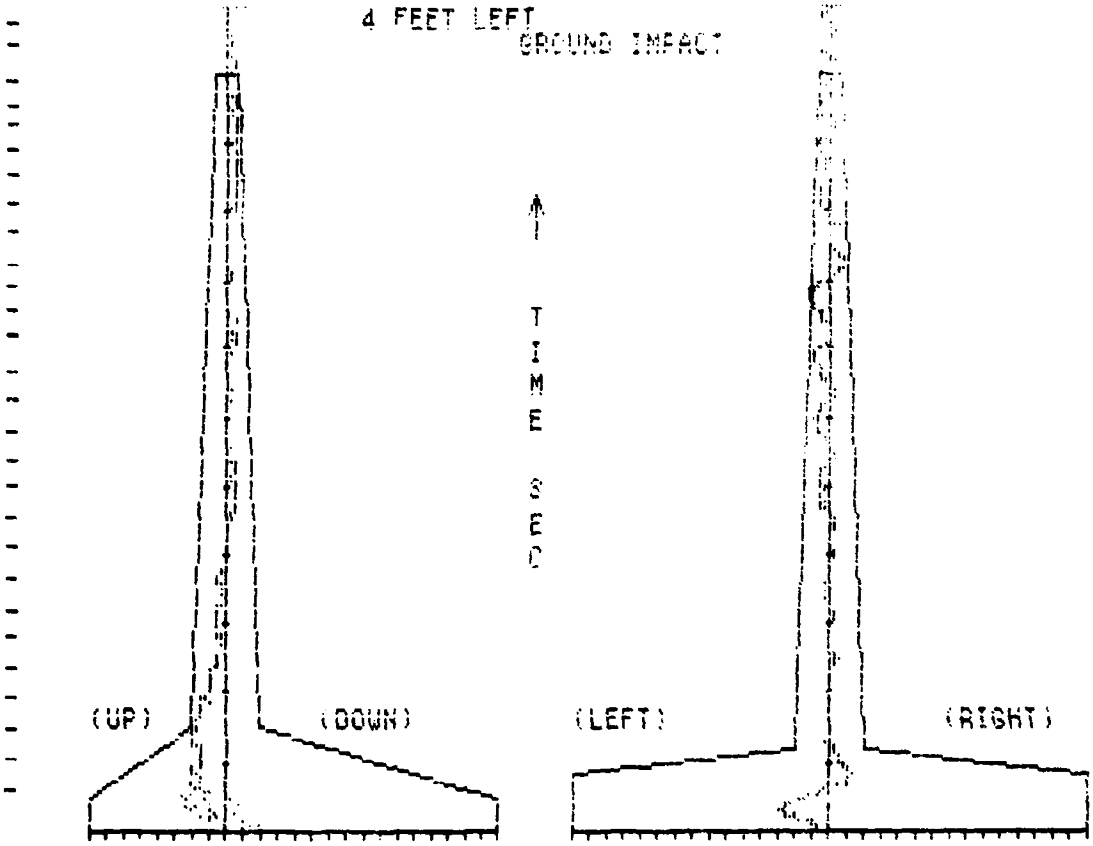


FIGURE II-2 (3 of 3) Missile Location vs Time (Vertical Plane) -HIT   GAE vs Time (Vertical Plane)



THRUSTERS: 27I/30A

FIGURE II-3 (1 of 3) GAE vs Time

(Vertical & Horizontal Planes)

-MISS

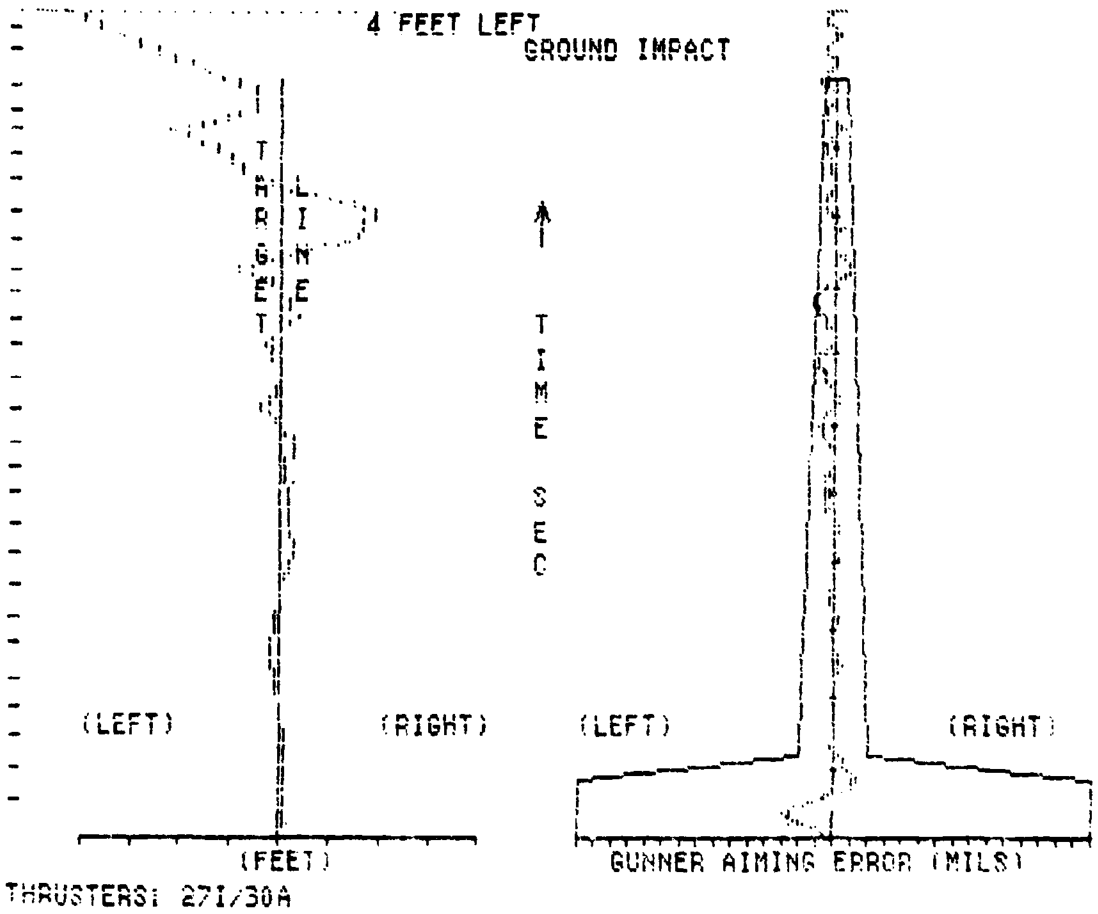


FIGURE II-3 (2 of 3) Missile Location vs Time (Horizontal Plane) GAE vs Time (Horizontal Plane)

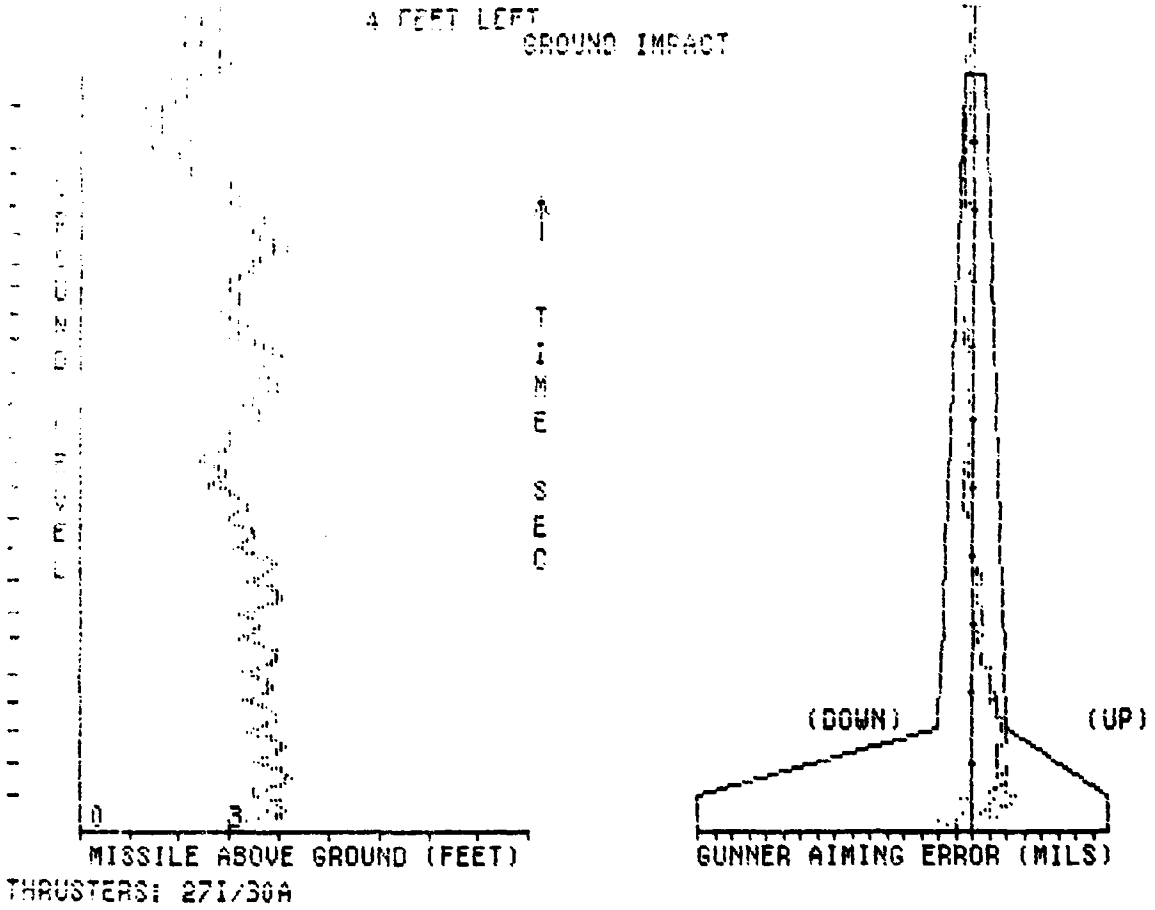


FIGURE II-3 (3 of 3) Missile Location vs Time (Vertical Plane) - MISS GAE vs Time (Vertical Plane)

SECTION III

SYSTEM DESIGN

A. Electro-Optics Subsystem

In order to solve the DRAGON flight equations several input parameters are required. These parameters are:

(a) trigger pull

(b) target position and range

(c) gunner aiming error

The STAGS measures the gunner aiming error with respect to a selected aim point on the miniature tank target, using the electro-optics subsystem.

Gunner aiming errors are determined by sensing the IRED on the target using a 100 x 100 matrix solid-state camera. The IRED is invisible to the human eye.

Solid-state imaging cameras are functionally similar to videcon type TV cameras, but with the added advantages of greater geometric accuracy, extended spectral range, higher sensitivity and scan rates, digital output, small size, low voltage and power requirements and the ruggedness and reliability of solid-state design.

The sensor is a solid-state photodiode array matrix having 10,000 pixels (100 x 100). The choice of lens determines the field viewed by the matrix camera. Using a 125 mm focal length lens and a target model distance of 22 feet, we have a field of view (FOV), of 1.05 feet or 48 milliradians. This FOV will accommodate the maximum excursions allowed for DRAGON, i.e., 32 mr horizontal and 22 mr vertical.

For a 1.05 ft FOV one pixel represents 0.126 inches on the terrain board.

Since the array is square the lengths in the X and Y axes are identical. The magnification, M, of the camera is the ratio of the FOV to the length of the array:

$$M = \frac{\text{FOV}}{\text{Array Length}}$$

where the array length is = 0.24 in. (0.60 cm total width/height) in both X and Y.

$$M = \frac{1.05 \times 12}{0.24} = 52.5$$

The static resolution is the array element spacing imaged into the object plane.

Resolution = Magnification x element spacing

Resolution = 52.5 x .0024 in. = 0.126 inches

This is equivalent to ± 7.5 in. resolution on a real world tank at the scaled range of 2,640 feet.

If a longer focal length lens is used the FOV is decreased and the resolution is improved.

Accuracy also depends on: image sharpness, contrast, vibration or movement of the object, light level and threshold setting of the camera.

The camera used is blemish free.

An IRED is located on the scaled model target and the center of the IRED's energy is calculated to determine hit location.

Because the IRED produces uniform illumination, the threshold setting on the camera can be adjusted to a fixed level, thus eliminating background interference.

The data from the photodiode array is electronically scanned to produce a sampled-and-held video output signal. The amplitude of each pixel is proportional to the incident light intensity integrated over the interval of one frame period. The camera essentially detects light to dark transitions of the digital area. The scene present on the camera is a light circle on a dark background. Transition data from the camera, stored as a digital line-by-line picture of the array, is handled by an interface unit. The DRAGON Flight Simulator Processor determines the GAE, from the transition data.

The electro-optic subsystem consists of the following equipments. (See Figure III-1)

Reticon MC 520 Camera

Reticon RS 520 Controller

Reticon RSB-6020 Interface Board

Nikon Zoom Lens (set at 125mm)

The RSB-6020 directly couples Reticon imaging cameras to Intel SBC/Multibus systems.

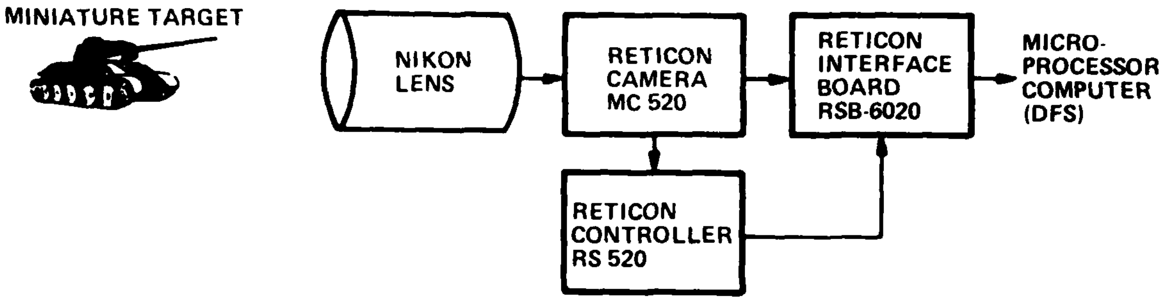

MINIATURE TARGET
NIKON LENS
RETICON CAMERA MC 520
RETICON INTERFACE BOARD RSB-6020
MICRO-PROCESSOR COMPUTER (DFS)
RETICON CONTROLLER RS 520

FIGURE III-1 ELECTRO-OPTIC SUBSYSTEM BLOCK DIAGRAM

B. Multiprocessor Subsystem

The multiprocessor subsystem includes six units with five being housed in the system chassis. The principal function of each of the separate units is:

(a) Personnel Interface Processing (PIP)

(b) DRAGON Flight Simulation (DFS)

(c) Sound Generation (SG)

(d) Target Control (TC)

(e) TV Display (TVD)

(f) Photodiode Array Processing (PAP)

System I/O is processed by the PIP, which is covered in the Computer Graphics and Video Subsystem Section.

Target control is detailed in the section Miniature Target Board.

The present section provides a description of the DRAGON Flight Simulator and the Photodiode Array Processor.

1. DRAGON Flight Simulator

The McDonnell Douglas Astronautics Company, Titusville Division, under Contract N61339-80-M-3518 provided a set of simplified equations and a computer program that approximate the DRAGON missile flight as directed by the gunner. (See Appendix A)

Six-degree-of-freedom equations are required to express the complete missile dynamics. Solutions of such equations were examined and simplied as much as possible by McDonnell while still maintaining a statistically accurate representation of weapon performance. Some of the simplifying assumptions were:

(1) Missile dynamics should be represented by a point mass solution,

(2) Small angle approximations to be used,

(3) The effect of tracker sampling on missile trajectory while in the linear field of view may be neglected.

The six-degree-of-freedom equations thus modified were exercised and compared to results obtained from the complete equations of motion. Modification to the thrust level and guidance parameters were made to tailor the trajectory to the more exact results. Sufficient comparative analysis was conducted to assure that the simplified equations gave acceptable results over a range of crossing and stationary target conditions and with a variety of gunner aiming errors.

Figure III-2 is the DRAGON simulation block diagram. The variables correspond with those of Figure III-3 which defines the important horizontal angles. These, and a similar set of vertical angles, were used in the McDonnel "BASIC" program which iterates the differential equations of motion using a "Delta Time" of 20 milliseconds. Thus a 10 second missile flight requires the generation of 500 solutions of the equations of motion.

The BASIC program was rewritten for an Intel Microprocessor Development (MDS) System (see Appendix A). The resulting program, while able to reproduce the McDonnell results, required several minutes to complete the 500 solutions for a simulated 10 second missile flight. It was, therefore, unsuitable for real time training.

An investigation of other floating-point-math techniques usable with Intel SBC-86/12, 8086, computers showed that real-time solutions of the missile flight could not be accomplished without using an 8087 coprocessor. The unavailability of the 8087 at that time made it ncessary to abandon the convenience of FP-math and recast the equations using integer arithmetic. This required close attention to the choice of suitable units for the variables because of the limited range of integer numbers: (-32,767, +32,767). Down-range distances, for example, are expressed in 2-inch units; 1000 meters (39,370 inches) being considered to be 19,685 "Down-range" units. Cross-range units are 0.05 inches for distances and 0.1 milliradians for angles. Unit selection for all variables is a compromise between the conflicting requirements of the desire to display variables over a wide range and the need to reduce quantization distortion while not exceeding the allowable integer range. Many comparisons between the integer and BASIC

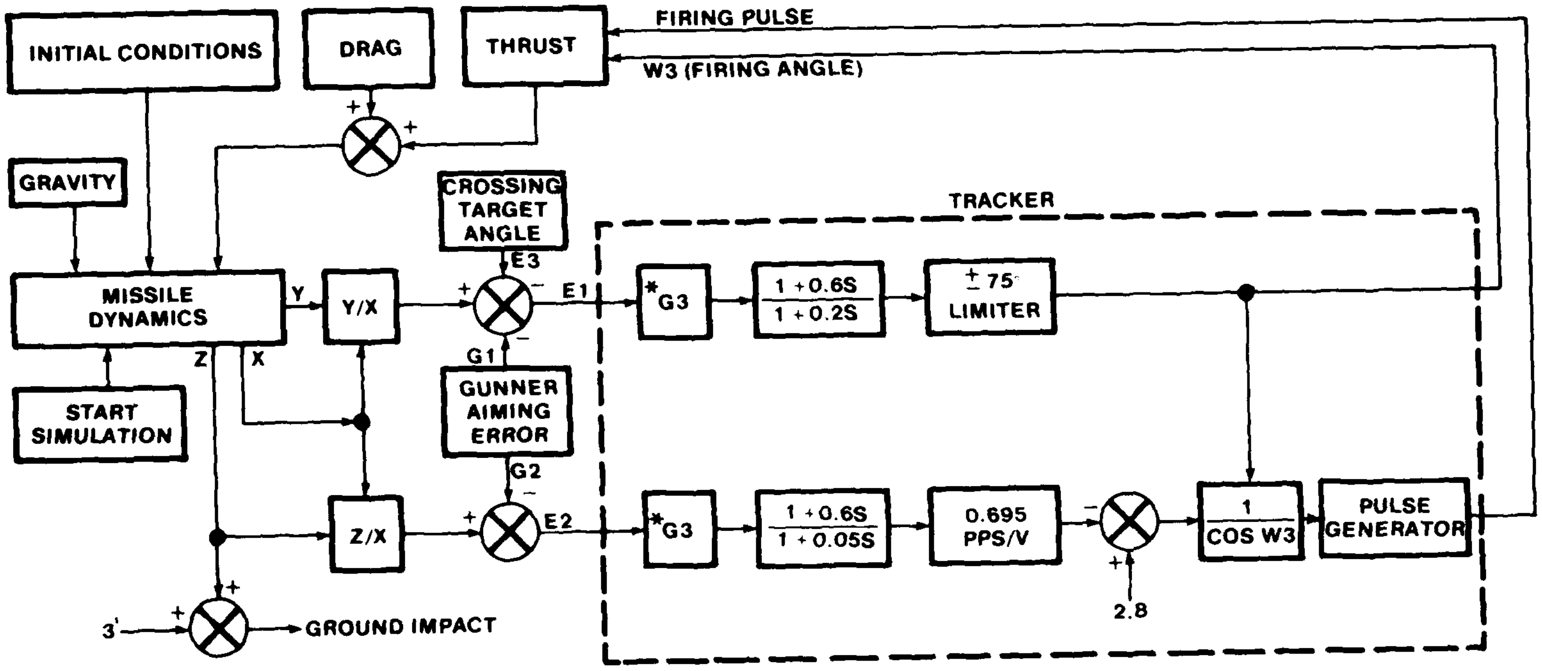


FIGURE III-2 SIMULATION BLOCK DIAGRAM

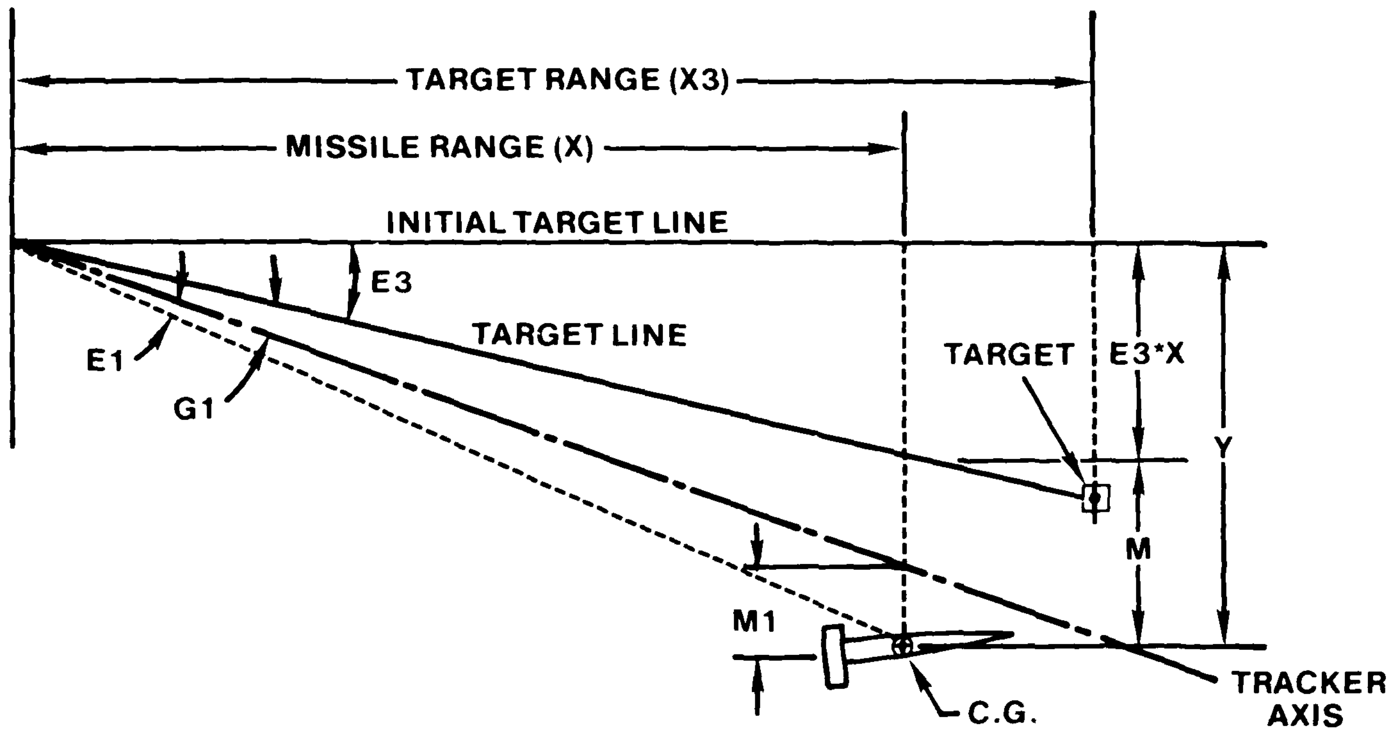


FIGURE III-3 HORIZONTAL PLANE GEOMETRY

program results have verified that good approximations to the DRAGON Flight characteristics are provided using integer arithmetic. Comments by experienced DRAGON gunners also support the validity of the approximations.

The DRAGON Flight Simulation Program includes five modules:

(1) Main-DRAGON-Module: A "Driver" module which calls other modules.

(2) DRAGON-Utility: Includes a number of start-up and other general procedures.

(3) DRAGON Flight Module: Includes the integer math missile dynamics, provides missile location information to the PIP, stores location data for possible reprise, and does the initialization of flight variables.

(4) DRAGON IR: Analyses the IR-spot data array provided by the following module.

(5) DRAGON XF: Transfers line-by-line data provided by the photo-detector line array processor into a complete picture array.

The first three modules are written in PLM 86; an Intel high level programming language. The last two are in 8086 assembly language. Total program code require slightly over 4K of ROM memory. Variable memory requires about 1K of RAM. The above programs are in Appendix B.

As noted previously, the program is located on an Intel SBC 86/12 board. This board, along with four others are housed in an Intel SBC 86/12 system chassis which provides eight card slots, power supply and ventilation. Cards within the chassis can communicate via the multibus motherboard. An SBC 86/12 provides dual-port RAM which can be accessed by both the on-and-off board processors. Missile position data resulting from the solution of the missile equations of motion are transferred to the PIP via the multibus for further processing and output. Data status bits are also read and written across the multibus as required.

Target motion is provided by the TC unit as described in the section on Miniature Modelboard. It is programmed via a stepper motor controller into which the desired target maneuver is input from a suitable menu located in program memory of the PIP. Target information needed by the DFS is transferred from the TC via the multibus.

The DFS also provides control signals to the sound generator for side-thruster pops, launch and impact explosions. It also provides signals for weight loss in response to trigger pull.

2. Photodiode Array Processor

Line scan data from the 100 x 100 photodiode array are initially stored in a set of ping-pong memories on a Reticon RSB 6020 board housed within the system chassis and attached to the multibus. Data are alternately read into ping or pong memory under control of a clock located within the Reticon RS 520. Data within the memory units give the location of light level transitions and indicates light-dark or dark-light transitions. The stored data also indicates when the last scan line is read.

After initialization, a last-line flag is output across the multibus to the DFS whicn causes the DRAGON XF program to begin the transfer of data from each line of the next 100 x 100 photodiode array frame to the SBC 86/12. The data read-out is then halted by the next occurrence of the last-line flag. The next 100 x 100 frame data are ignored during the analysis of the transferred frame data. New frame data analysis results are provided every other frame.

The frame rate of the Reticon camera is 100 frames per second so new IR-spot position data are provided 50 times a second or with a 20 millisecond period. Occurrence of the last-line flag acts as the master system clock with all data processing starting with its assertion.

Figure III-4 illustrates the various devices controlled by the multiprocessor subsystem.

An automatic zeroing method has been incorporated to ease day to day boresighting of the gunner sight to the miniature targets. Zeroing is accomplished by supporting the DRAGON launcher on its resting stand and aiming the scope crosshairs at the desired aim point (Recoil and weight loss off). A Control B (CTRL B) is input from the console and the DRAGON trigger is squeezed. The computer will "read" the first frame of data from the Reticon 100 x 100 matrix camera and use this aim point for its reference boresight.

Subsequent firing will use the new boresight until the zeroing procedure is repeated or the computer is turned off.

C. Computer Graphics and Video Systems

The DRAGON computer graphic visual presentation is prepared by the Personnel Interface Processor. In addition to this processor a computer graphics board, a phase-locked-loop sync board, and an EIA composite sync generator are used. Figure III-5 shows the complete graphics and the video subsystem.

Computer generated graphics nrovide two major functions:

(1) Real-time video graphics are generated for the gunner sight. These graphics include a simulated missile which include thruster firings, smoke obscuration during initial launch and a final explosion.

(2) Real-time graphics are generated for the instructor which indicate both vertical and horizontal gunner aiming errors. Also, for follow up analysis, graphics may be presented for gunner aiming error versus time and missile position versus time.

Gunner's sight real-time computer graphics are generated on a Matrox 256 x 256 x 4 graphics board. Sixteen levels of gray scale provide for a full range of visual intensity which allows for smoke generation which varies from fully transparent to completely opaque in sixteen levels. The Matrox RGB-256 is a graphics imaging system in which a complete gray scale capability has been integrated onto a single printed circuit board. The card includes built-in NTSC (American) and PAL (European) gray scale encoder which can provide up to 16 shades of gray. The encoders permit the RGB-256 to directly drive standard

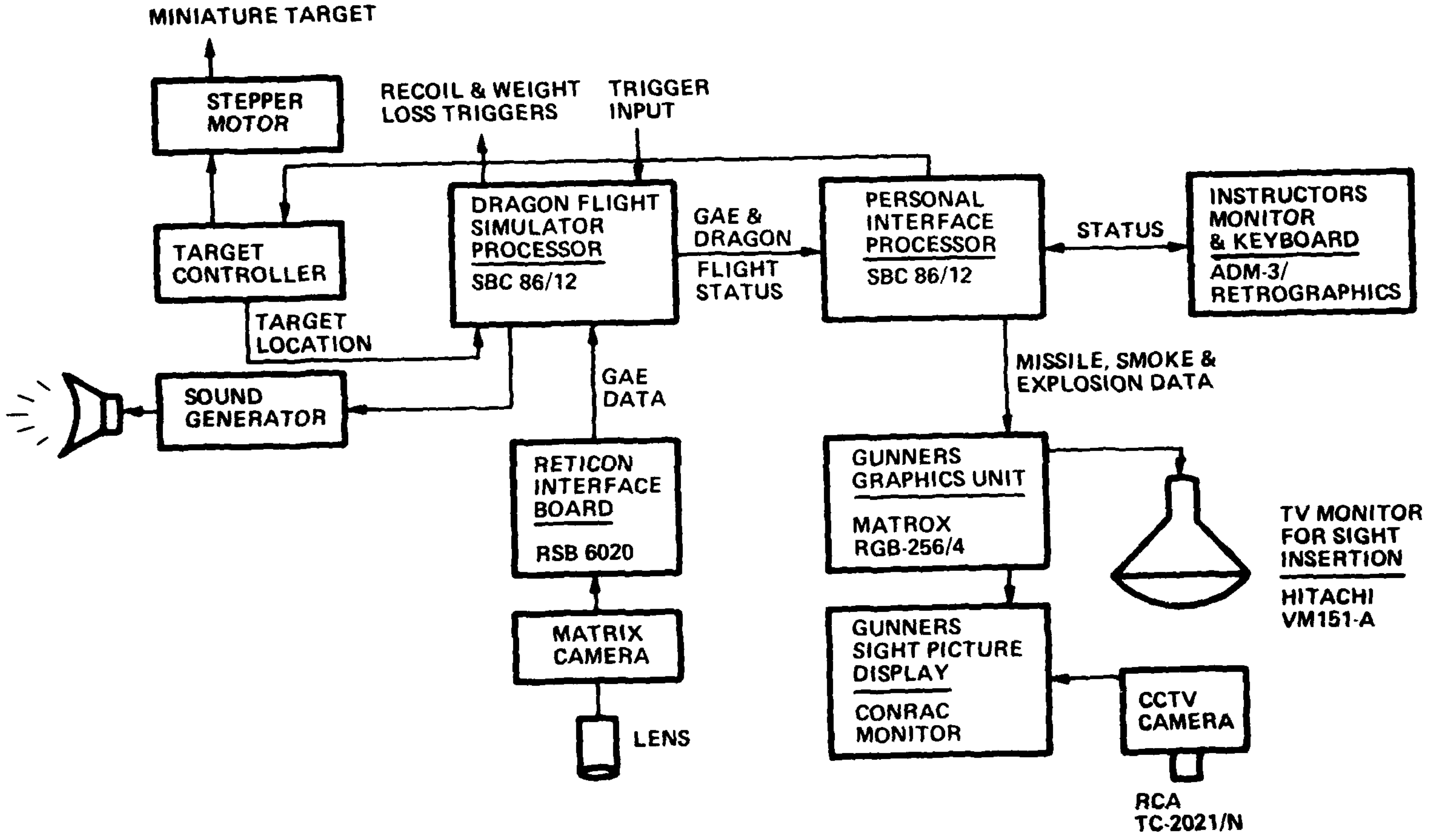


FIGURE III-4 MULTIPROCESSOR SUBSYSTEM

low cost black and white TV monitors on a single 75 ohm cable. It features the industry standard Intel Multibus which makes it directly plug compatible with all Intel single board computers.

Real-time video graphics are generated by the Personnel Interface Processor (PIP). The PIP receives gunner aiming error information from the DRAGON Flight Simulator (DFS) as well as missile angle from the line of sight of the gunner to the missile position (angle El from Figure III-3). The gunner aiming error is used to position the final explosion (hit or miss) in the DRAGON sight. The angle El is used to position the missile in the DRAGON sight.

Gunner Graphics:

DRAGON sight graphic missile simulation is accomplished by first deriving the missile position from the Angle El. Second, the size of the missile is determined by the elapsed time since the missile launch. Third, the brilliance of the missile is determined by the elapsed time since launch and if a thruster is being fired.

The size of the missile shrinks from 10 pixels down to 1 pixel from launch to maximum range. The brilliance decays from a level of ten (with fifteen being most brilliant) to a level of zero at minimum range.

An octagon was selected as the simulated missile shape as this can be quickly calculated for real-time graphics. This shape appears mostly as a circular area to the DRAGON gunner.

Smoke is simulated in the DRAGON sight by modulating the background level, i.e., overall gray scale setting of the entire graphic video insertion in the gunner sight. It is possible to tell the RGB-256 graphics board to "erase" to any given gray scale level between zero and fifteen, with zero being black (transparent) in the gunner sight and fifteen being white (opaque). The levels of background are modulated with time to effect a smoke simulation. A typical smoke simulation might consist of starting from level zero rising to level fifteen, dropping to level eight, back up to fifteen, down to four, up to eight and down to zero during a period of one to two seconds.

The final explosion of the missile and/or tank is simulated at the end of the DRAGON flight and inserted, via the RGB-256 graphics board, into the gunner sight. The explosion is a series of geometric star shapes indicating either a hit or miss. The PIP uses the missile-to-aim-point information to position the explosion wherever the missile was as it impacted the target or ground. A ground explosion is similar to a target explosion; however, it differs by only exploding in an upward sense. Thus the DRAGON gunner has visual feedback through his sight indicating hit and miss. The computer generated graphics are passed directly to the gunner's sight through a Hitachi VM151A, one and a quarter inch, closed circuit television (CCTV) monitor. The optical arrangement is shown in Figure III-6. The television screen appears at infinity along with the viewed scene through the 6x scope. The CCTV is mounted inside the DRAGON IR tracker housing and electronics for the CCTV are located where the IR tracker electronics are normally located at the bottom of the tracking head.

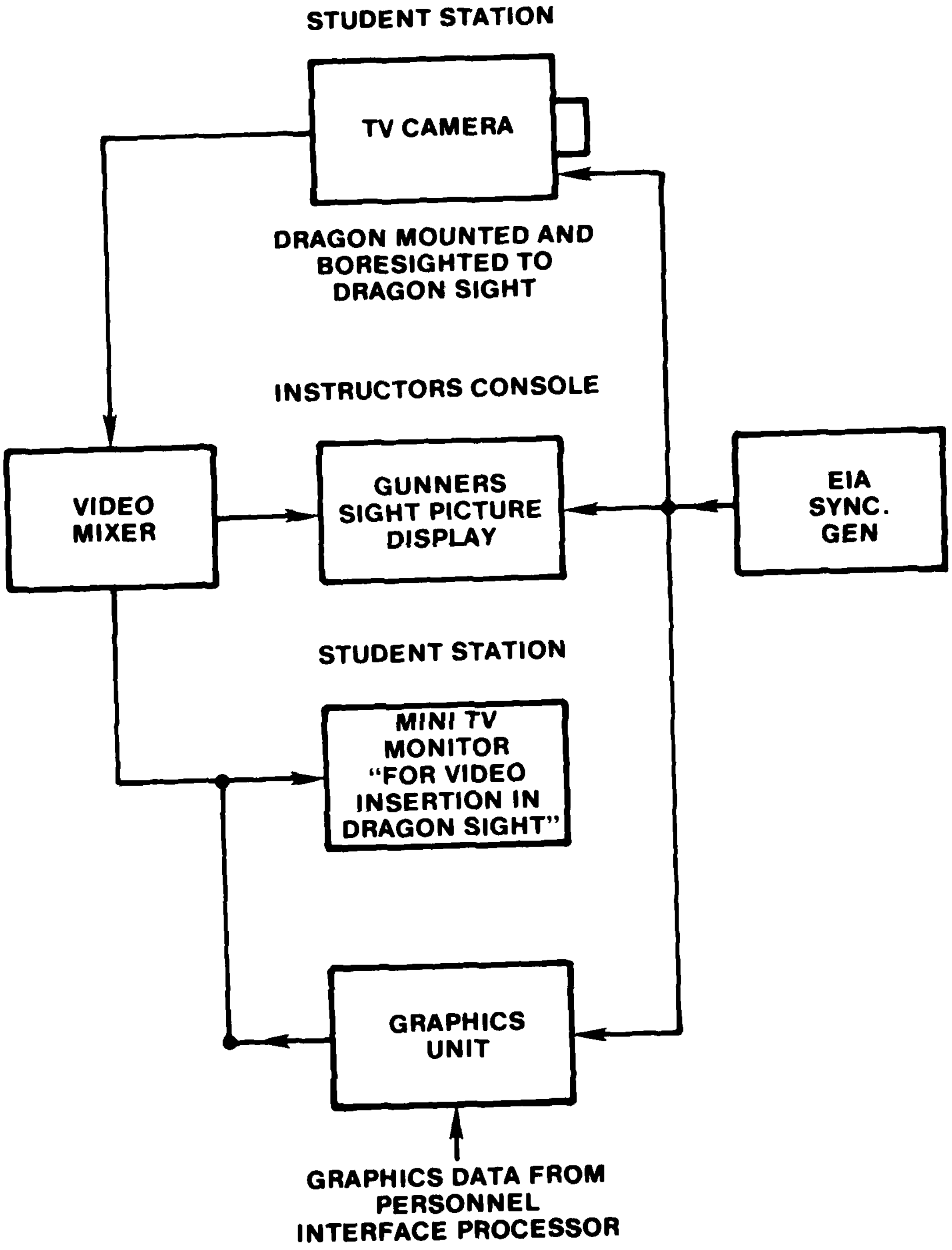

STUDENT STATION
TV CAMERA
DRAGON MOUNTED AND BORESIGHTED TO DRAGON SIGHT
INSTRUCTORS CONSOLE
VIDEO MIXER
GUNNERS SIGHT PICTURE DISPLAY
EIA SYNC. GEN
STUDENT STATION
MINI TV MONITOR "FOR VIDEO INSERTION IN DRAGON SIGHT"
GRAPHICS UNIT
GRAPHICS DATA FROM PERSONNEL INTERFACE PROCESSOR

FIGURE III-5 COMPUTER GRAPHICS AND VIDEO SUBSYSTEM

Instructor Console Graphics

The instructor console graphics subsystem is composed of two units, a television representation of the gunner's sight picture and a graphical plot of gunner aiming error versus time and/or gunner aiming error versus missile position.

The television representation of the gunner's sight is accomplished by mixing the gunner's sight TV camera, which is boresighted to the 6x gunner sight, with the video graphics presented to the gunner's sight. The composite picture presents to the instructor an image of the gunner's sight which includes the target, rocket, smoke, crosshairs and final explosion.

The instructor's television representation of the DRAGON gunner's sight picture and computer graphic missile, smoke, crosshairs and explosion simulation is combined from three sources.

The model board target and terrain is looked at by a closed circuit television (CCTV) camera which is mounted outboard on the DRAGON launcher tube. This camera is zeroed to the gunners 6x scope and has the same field of view as the scope. The camera used is an RCA TC-2021/N with a NUVICON camera tube and a 135mm f3.5 still camera lens. This camera was chosen for its small size and low weight.

Secondly, a video mixer combines the CCTV image with the Matrox RGB-256 computer graphics. The combination of CCTV video and computer graphics is then a representative visual image of the gunner sight picture except for the crosshairs.

Crosshairs are added, to complete the instructor sight picture display, by passing the video presentation through an electronic crosshair generator. The crosshairs are adjustable in position and width.

The graphical plot of the gunner aiming error (GAE) versus time for both horizontal and vertical error are presented in real-time during the missile flight. The graphs indicate the actual gunner aiming error during the flight as well as the limits for a 95% probability of hit performance. The guidance rocket thruster firings are shown when they are fired as well as a final actual count of the thrusters fired versus the ideal number of thrusters that would have been fired for a given target distance with perfect aim. At the end of a flight, displayed results show the miss distance, in feet, where the missile passed the target. If the missile struck the ground before passing the target, a message is displayed stating "ground impact" as well as the remaining distance to the target when grounded. If a hit is scored a hit message is displayed to mark the event.

After a missile flight a reprise of the flight may be called. A horizontal reprise replays the horizontal GAE and the horizontal missile position versus time. Likewise the vertical reprise replays the vertical GAE and the vertical missile position versus time. The reprises indicate all the hit/miss summaries of the first real-time plot.

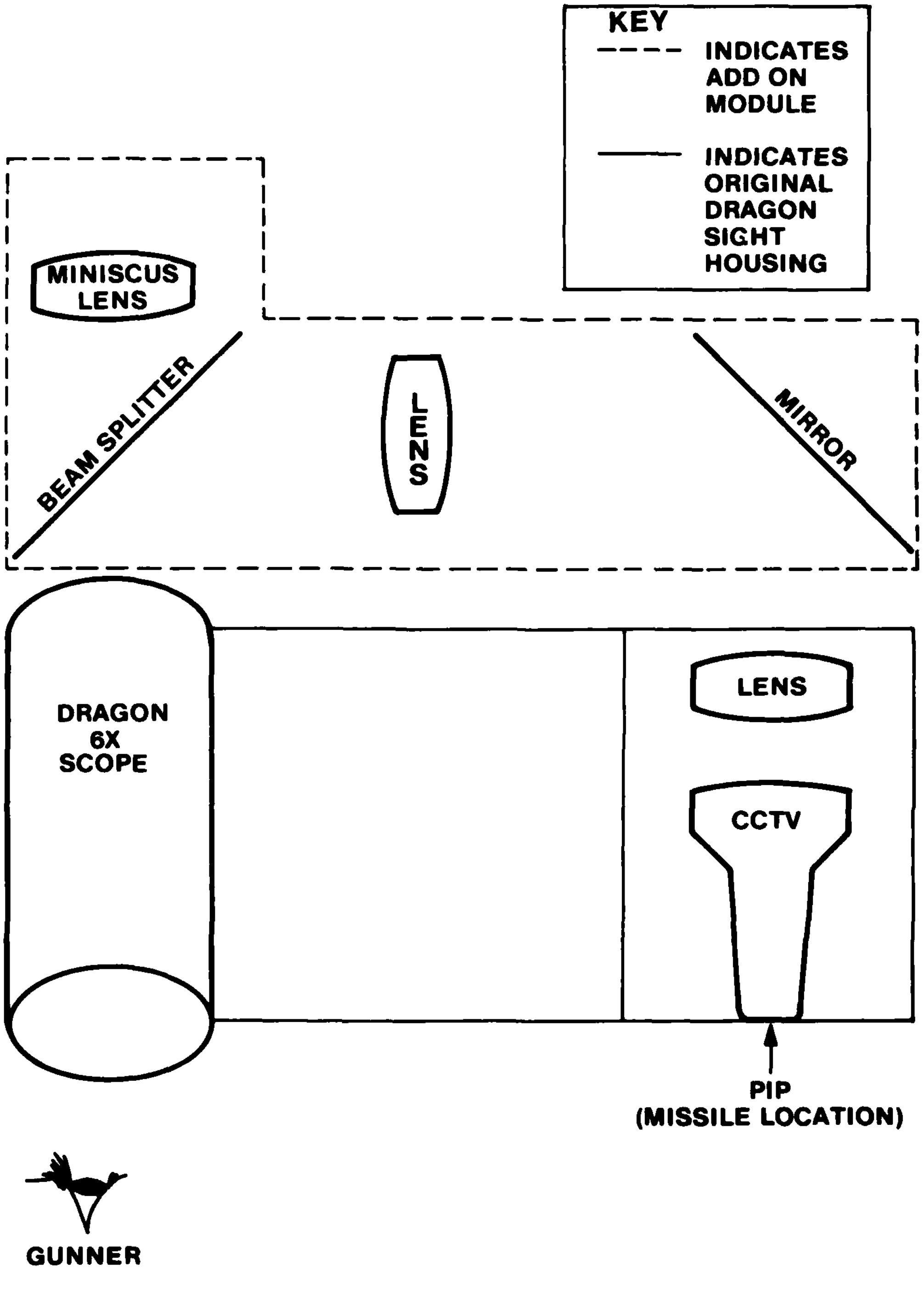


FIGURE III-6 DRAGON GUNNER'S SIGHT SYSTEM

Any of the computer graphic plots may be made into a hara-copy printout. The hard-copy may include the gunner's name or other pertinent data as desired bv the instructor.

The instructor's diagnostic graphs, keyboard controls and hard copy printouts are controlled by the PIP through an ADM-3 dumb terminal and Digital Engineering Retrographics RG-512 graphic board and Digital Engineering GP-100 hard copy printer.

The operation of the Retro-Graphics equipped ADM-3A can be best understood by considering the RG-512 card as the terminal controller and the ADM-3A as a "peripheral" device. The RG-512 is situated in series between the ADM-3A and the serial input to the terminal. This means that all incoming ASCII will be received by the RG-512 and processed. Input to the terminal will only reach the ADM-3A circuitry if it is transmitted there through the RG-512.

The RG-512 can perform several functions on the incoming data. The function performed depends on the actual ASCII code received and the RG-512 operating mode. Data may be retransmitted to the ADM-3A as mentioned above if the data were alphanumeric text. This is the usual function performed by the RG-512 when in the ADM-3A Alpha Mode. Certain control codes, called mode transition codes, can set the RG-512 to one of the two graphics modes, the Vector Mode or the Point Mode. An additional alphanumerics mode, the 4010 Alpha Mode, is included and can also be entered by sending the terminal the appropriate mode transition code.

After entering one of the graphics modes, subsequent input it interpreted as x-y coordinate data and is used in the generation of a point or vector display. The RG-512 does not retransmit an ASCII code to the ADM-3A if it is being used as an x-y coordinate.

The RG-512 employes the "bit map" method of storing graphic images. This means the information is stored in a digital memory as a rectangular array of bits. Each bit in this memory is mapped onto the CRT screen and can cause a bright point to be displayed. The RG-512 displays graphs and pictures by writing the proper bits into the graphics memory. This architecture has several advantages over the traditional storage tube approach which has dominated lower cost graphics terminal designs. Since the CRT is not relied upon for storage of the image, less expensive CRTs employing more conventional long life, brighter phosphors can be used. Another important by-product is the ability to selectively erase portions of the screen. This is desirable if the application requires the use of dynamic displays employing motion or rotation to convey information.

All circuitry for the RG-512 is packaged on a single 12" x 12.31" printed circuit card. This circuitry consists of four functional elements: Z-80A microprocessor and control, 128,000 bit graphics RAM, raster synchronization and CRT refresh, and power supply.

The Z-80A microprocessor and control section performs command decoding and is responsible for the writing of information into the graphics RAM. The Z-80A automatically generates vectors from transmitted endpoints and also performs scaling and character generating functions.

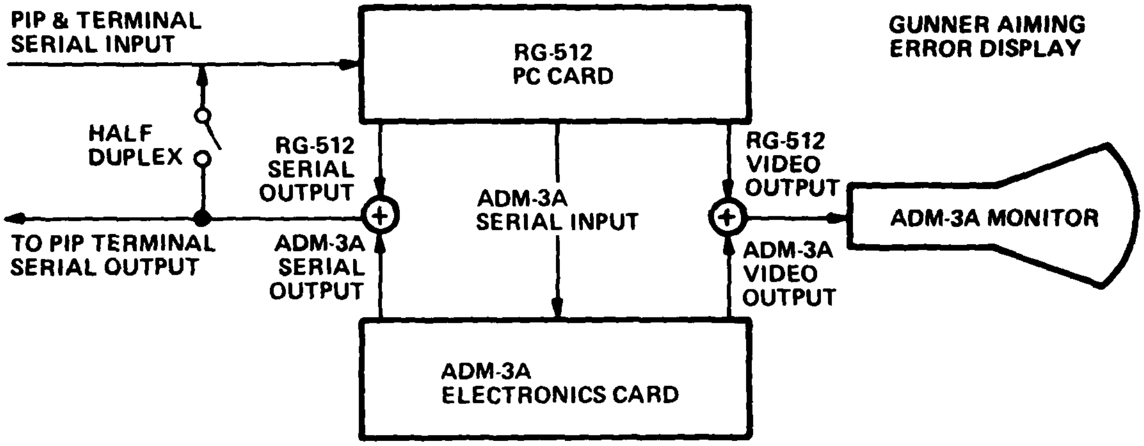


FIGURE III-7 RG-512 FUNCTIONAL BLOCK DIAGRAM

The GP-100 Graphics-Printer is designed to interface with the Retro-Graphics upgraded Lear Siegler ADM-3A (or 3A+) computer terminal.

As an optional feature to Retro-Graphics, Graphx-Printer reproduces the terminal's displayed graphics (or alphanumerics) on hardcopy. The electrosensitive device is ideal in those computer graphics applications where clear, economical printing is needed.

The GP-100's printing system is electrosensitive. This advanced technology allows the GP-100 to quietly print the displayed graphics within 16 seconds and alphanumerics at 170 lines per minute.

With the electrosensitive printing system all graphics and alphanumerics are drawn by a durable printhead that produces dark, clear images with no distortion. In other words, the high resolution graphics screen image is reproduced, dot-for-dot, on hardcopy.

Additionally, paper is the printer's only consumable. Typically an 8 1/2" x 11" hardcopy costs less than 3¢. Also, the paper can be photocopied, and it will not fade or yellow with time.

The electrosensitive paper is made from a base of black coated ordinary bond paper deposited with a thin aluminum layer; this gives the unprinted paper a metallic appearance. During printing, an electric current flows through styli that touch the aluminum. This action vaporizes the aluminum at the point of contact and exposes the contrasting black under-layer to form the graphics or character image.

The print head consists of 12 styli mounted one above the other as a unit. This head is attached to a carriage which moves the styli from left to right and maintains them in contact with the paper. The styli are pulsed electrically while crossing the paper so that the required characters or graphics are printed. During printing, each stylus burns a single dot in response to a current pulse. At the end of the line, the carriage moves the head away from the paper and returns to the left hand margin without touching the paper.

D. Computer Generated Sound System

Simulation of sounds produced during an actual DRAGON missile firing is accomplished by interfacing an Intel 8748 microcomputer to a General Instruments AY-3-8910 Programmable Sound Generator (PSG). Data necessary for the PSG to reproduce sounds is acquired from the permanent memory of the microcomputer. During missile flight time the DFS processor simply selects the sound to be made and communicates its choice to the microcomputer. This approach allows the processor to handle sound-making decisions with minimum time take from its primary functions.

The choice of sounds available to the DFS processor are:

(1) Gyro wind-up

(2) Missile launch explosions

(3) Rocket thruster motor firing

(4) Target missed explosions

(5) Target hit explosions

The General Instruments Programmable Sound Generator (PSG) is a 40 pin, 8 bit device with microprocessor compatibility. The device features three independent analog channels each with access to its own tone generator. A 16 control register array communicates to the microcomputer through an eight bit bi-directional port. Four lines are alloted for bus control logic (read and write). Each tone generator looks to two registers within the array for a 12 bit tone period. A range of frequencies covering the full eight octaves of the equal tempered chromatic scale is available.

Pseudo-random noise may be mixed to any or all channels from a noise generator with basic frequencies of 4 KHz to 125 KHz. Two modes of output control are available for each channel. The fixed level amplitude mode selects an amplitude specified in the array by the microcomputer. For use in this system the variable amplitude mode is selected, forcing an envelope generator to control the shape and cycle of all outputs. Controlling the envelope generator is a 16 bit tone period within the array allowing for frequency ranges of 12 Hz to 7812.5 Hz and a five bit shape/cycle control register. Three D/A converters supply 0 to 1 volt signals to the output channels.

To accurately represent the flight of a DRAGON missile as it moves downrange two sound phenomena must be simulated:

(1) Time delay due to the differences in the speeds of light and sound and,

(2) The Logarithmic decay in the amplitude of sound with distance.

Software developed for the microcomputer closely approximates these conditions within a 1000 meter range.

As shown in Figure III-8, the outputs of the PSGs are input to circuits whose function is to control the amplitude of the sound. These circuits consist of operational amplifiers with closed loop gains under direct control of the microcomputer. An Intel 8243 1/0 port expander is used to select feedback networks of the operational amplifiers. Harris analog switches HI518 and HI304 under the direction of the 8243 provide a variable feedback network for the rocket thruster sounds. The hit and miss explosions pass through a separate operational amplifier circuit utilizing the same HI518 for gain control. To further massage the rocket thruster sounds a Pioneer Reverberation Amplifier is inserted between the output of the operational amplifier and a Bozak Pre-amp and Mixer. Rocket thruster sounds, hit and miss explosions, and the launch explosion all pass through the mixer and a Bozak Amplifier. The launch explosion is generated from a Frazier 8-ohm speaker located at the trainees station. Rocket thruster firings are heard from a speaker near the modelboard as well as hit/miss explosions. The gyro wind-up noise emanates from a speaker located within the DRAGON tube.

The DRAGON Flight Simulator processor initiates a timer within the microcomputer upon request of a launch explosion. Thereafter, each request for a sound by the processor causes the microcomputer to inspect the timer. Assuming the missile travels at an average speed of 280 feet per second the microcomputer is able to approximate the distance covered and determine the appropriate sound amplitude. The microcomputer selects one of thirteen levels of amplitude, for rocket thruster sounds, decreasing logarithmically from a gain of 10 to 1 over a time span of 11 seconds corresponding to a distance of 1000 meters. When the missile hits its target a series of three rapid explosions are generated, each explosion louder than the previous. For a missed target two explosions occur. Distance in terms of three ranges is the criteria used to determine the amplitudes of these explosions. These ranges are: low range (less than 333 meters), mid range (334 to 630 meters), and a high range (greater than 630 meters). The gyro wind-up noise and the launch explosion have fixed amplitudes.

Time delay associated with distance covered by the missile is accomplished upon inspection of the timer for each requested sound after launch. Before signals are passed to the PSG to create a sound, software completes a sequence of three delays. The first delay represents the real-time between requests from the processor. This timeout occurs only when two or more requests are made before

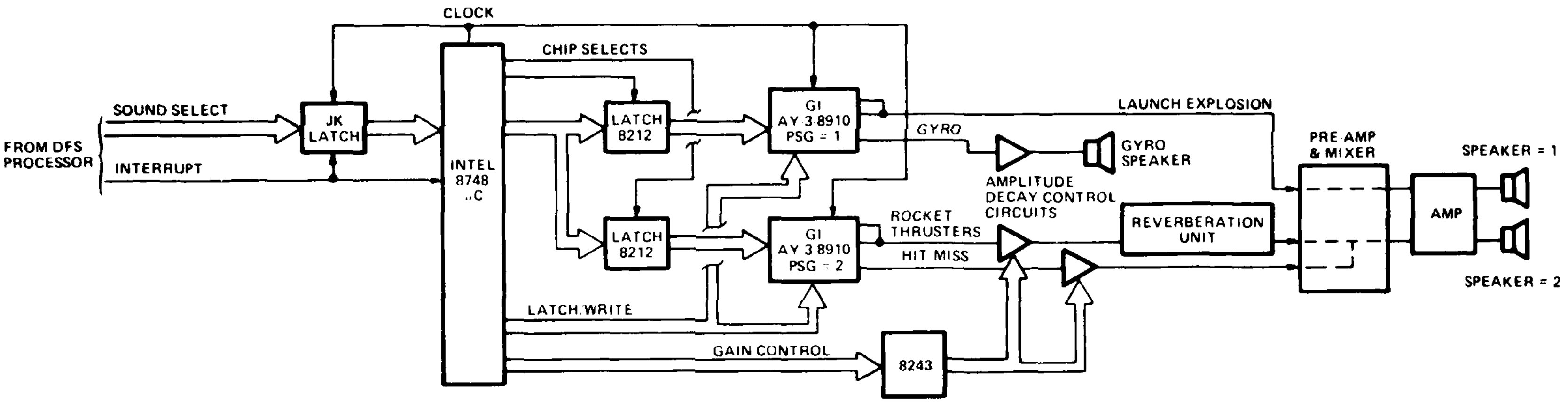
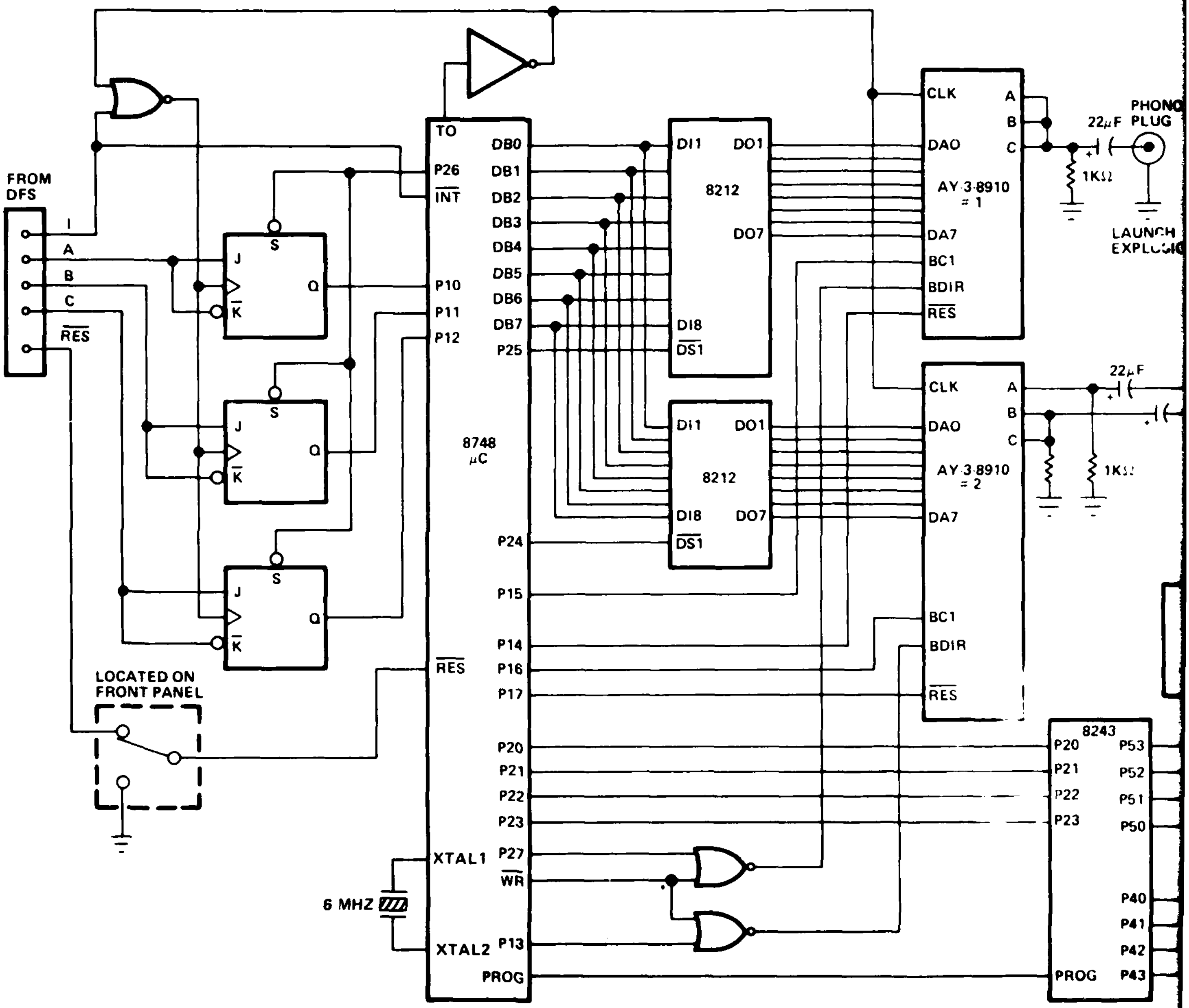


FIGURE III-8 SOUND SYSTEM FUNCTIONAL BLOCK DIAGRAM


FROM DFS
I
A
B
C
$\overline{RES}$
J
S
Q
$\overline{K}$
LOCATED ON FRONT PANEL
TO
P26
$\overline{INT}$
P10
P11
P12
$\overline{RES}$
XTAL1
XTAL2
6 MHZ
8748 μC
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
P25
P24
P15
P14
P16
P17
P20
P21
P22
P23
P27
$\overline{WR}$
P13
PROG
DI1
DO1
8212
DO7
DI8
$\overline{DS1}$
CLK
DAO
DA7
BC1
BDIR
$\overline{RES}$
AY·3·8910 = 1
AY·3·8910 = 2
A
B
C
22μF
1KΩ
PHONO PLUG
LAUNCH
EXPLOSIO
8243
P53
P52
P51
P50
P40
P41
P42
P43

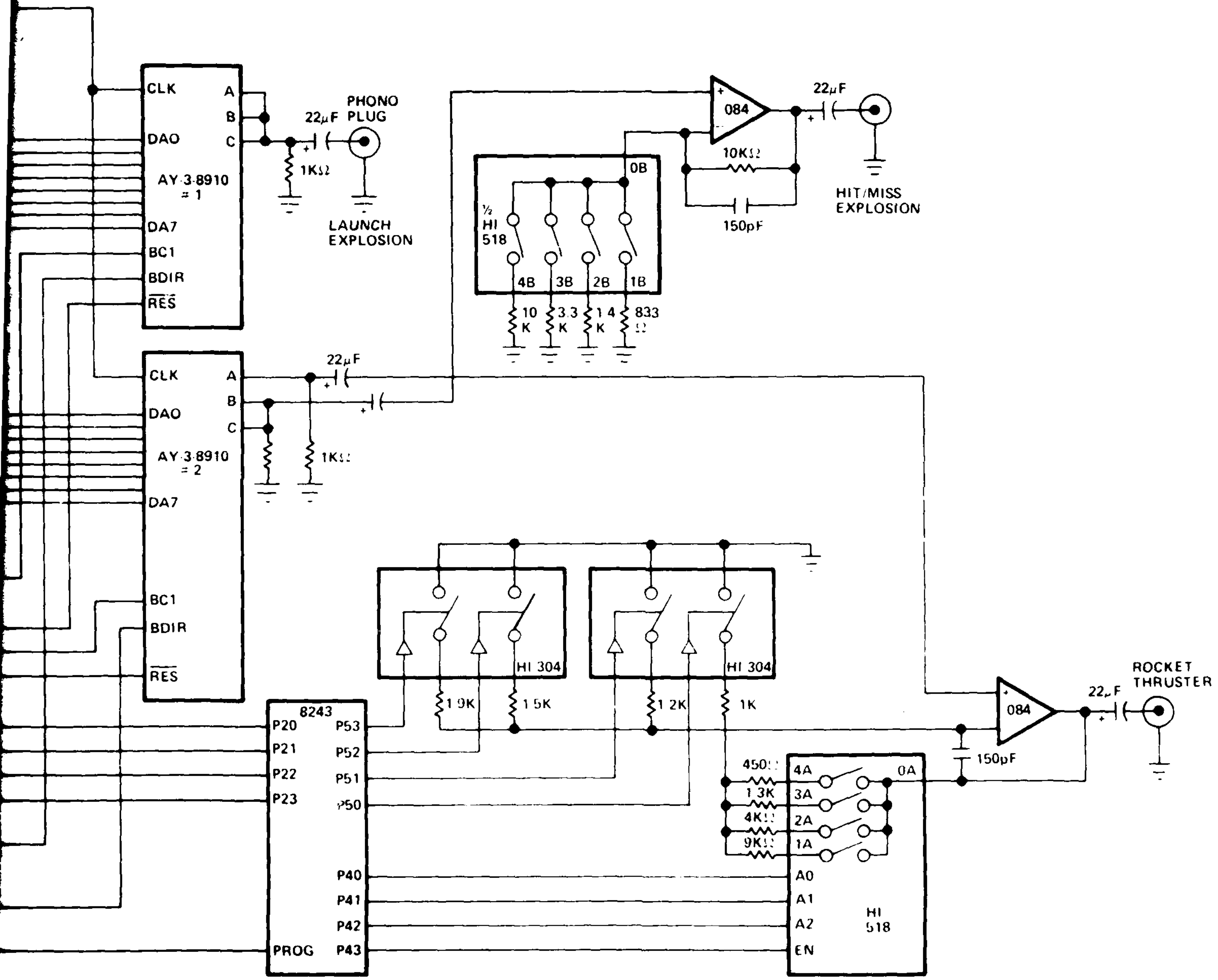


FIGURE III-9 SOUND SYSTEM SCHEMATIC

2

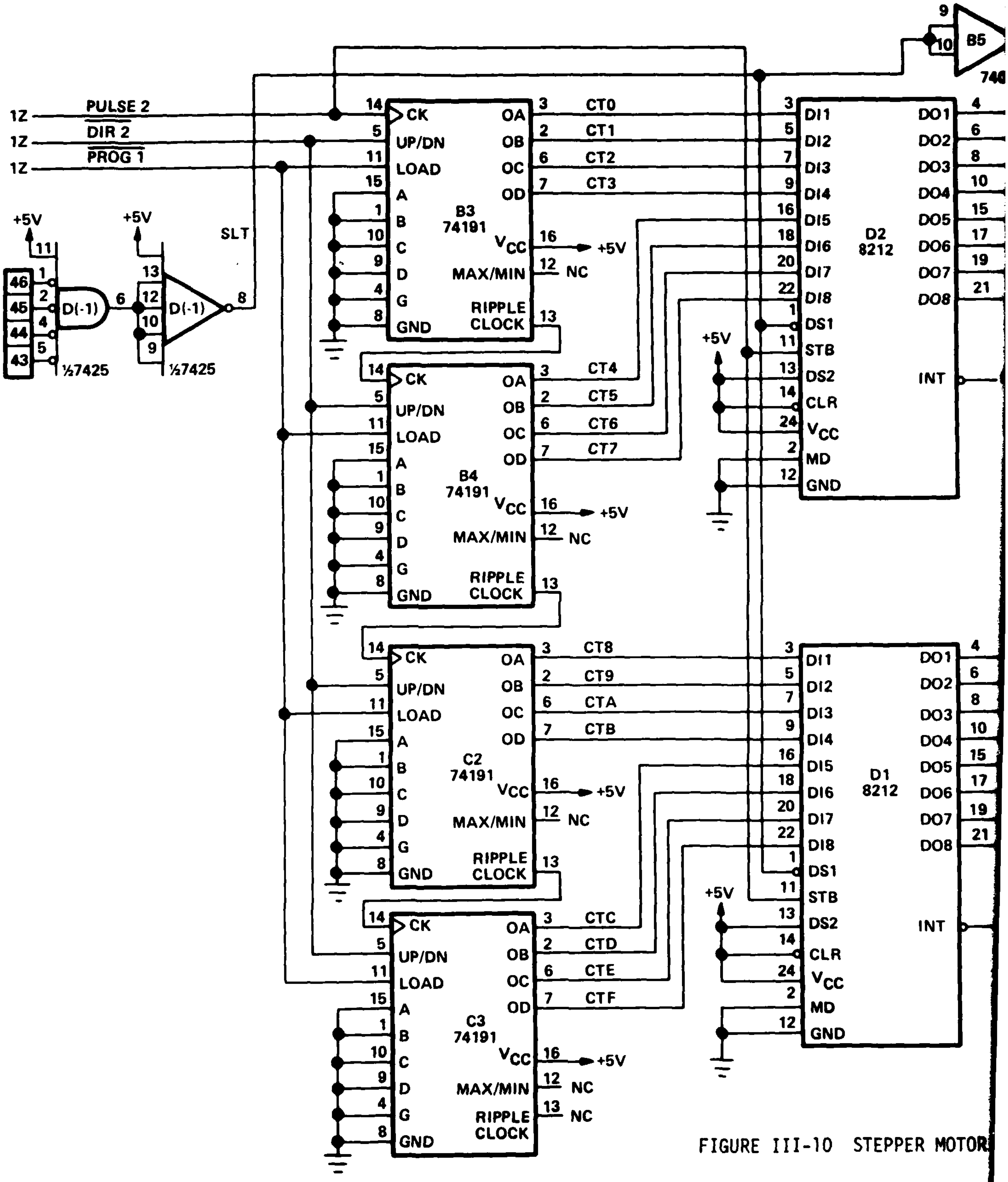


FIGURE III-10 STEPPER MOTOR

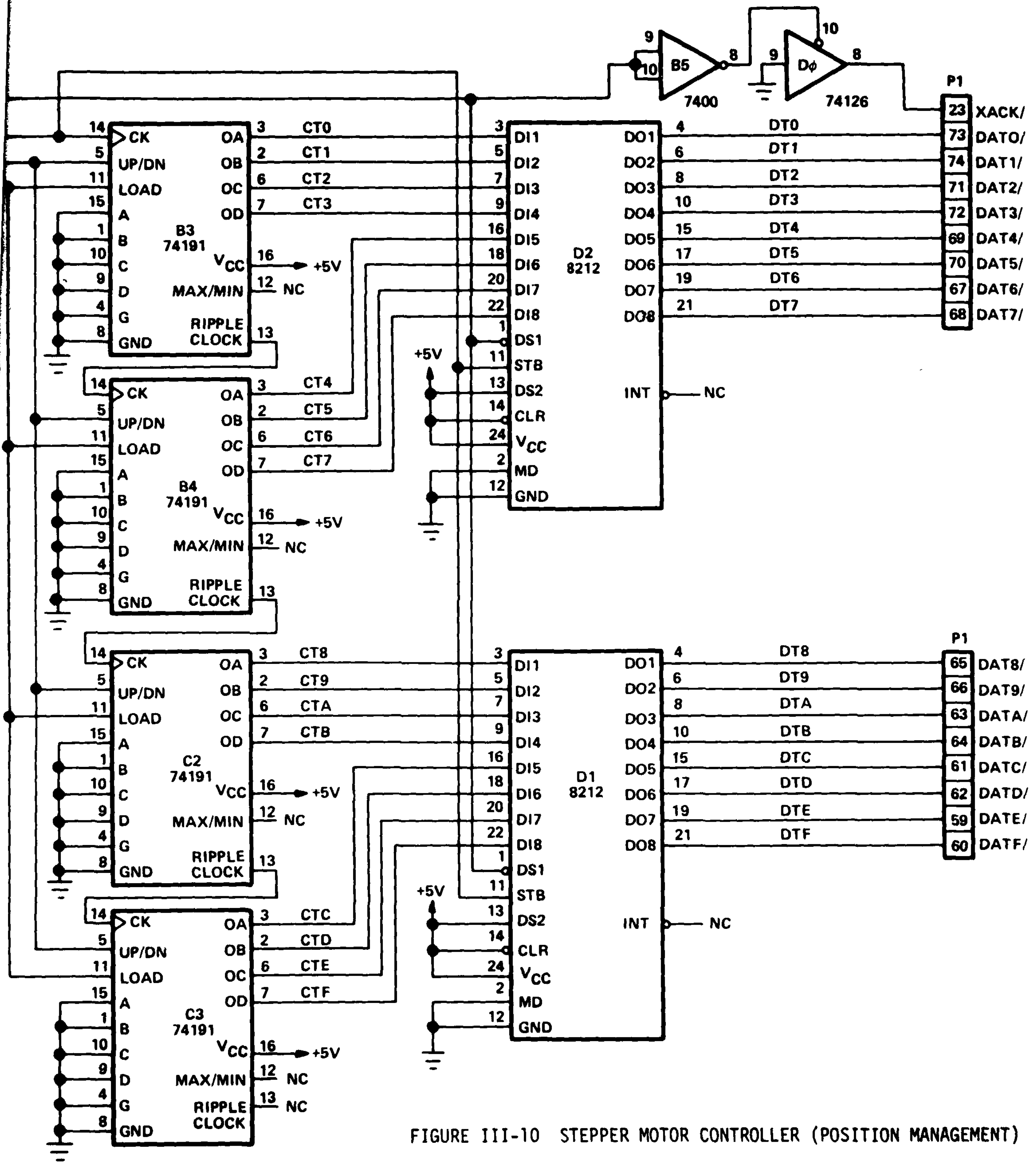


FIGURE III-10 STEPPER MOTOR CONTROLLER (POSITION MANAGEMENT)

2

the first request is serviced by passing signals to the PSG. The real-time between any two requests represents distance traveled by the missile and is decoded into the second time delay as determined by the time required for sound waves to travel this distance. These incremental time delays are accumulated in the microcomputers data memory. The third time delay before a sound is made is the cumulative total of all the second time delays that have already been decoded. The complete algorithm produces a series of logarithmically decaying, time delayed, sound waves that approximate the actual conditions within a 1000 meter range.

E. Miniature Target Board

Because most anti-armor devices use high power telescopes to view the targets, a 1/120 scaled miniature model was chosen. The target model has an IRED located at the center of the target mass. The model is moved using a stepper motor. The stepper motor controller is a stand-alone intelligent controller that is independent of the host computer, the Personnel Interface Processor, except for loading the scenario. The stepper motor controller uses a high level language for control of the stepper motor direction, position, speed and acceleration. The tank is moved over a 40 inch track. It takes 5,240 half steps to run this track. Using this system the tanks location is known to 0.0076 inches on the model board. Scaled to the real world one half step moves the tank 0.9 inches.

The controller utilized is a Cybernetic Micro Systems, CY 512. The CY 512 controller is a standard five volt, 40 pin LSI device configured to control a 4-phase stepper motor. The CY 512 interfaces to the microcomputer using parallel TTL input. It also has a software controllable pin which is used to initiate movement of the tanks turret.

Hi-level commands to control the device are stored externally in the PIP processor.

Under instructor control a scenario can be selected. The commands are then transferred and stored internally in a program buffer in the CY 512. The CY 512 outputs sequence the stepper drive circuits that consist of standard Darlington drivers. When absolute position commands are executed, the CY 512 automatically determines whether it is necessary to move CW or CCW to reach the specified target position.

Tank position is measured by a 16-bit counter consisting of four 74191 TTL chips. (See Figure III-10) The counter is reset whenever a new scenario is loaded into the CY 512. The counter then records half-steps of the stepper motor.

The CY 512 interface is shown in Figure III-11.

The Darlington drivers are shown in Figure III-12.

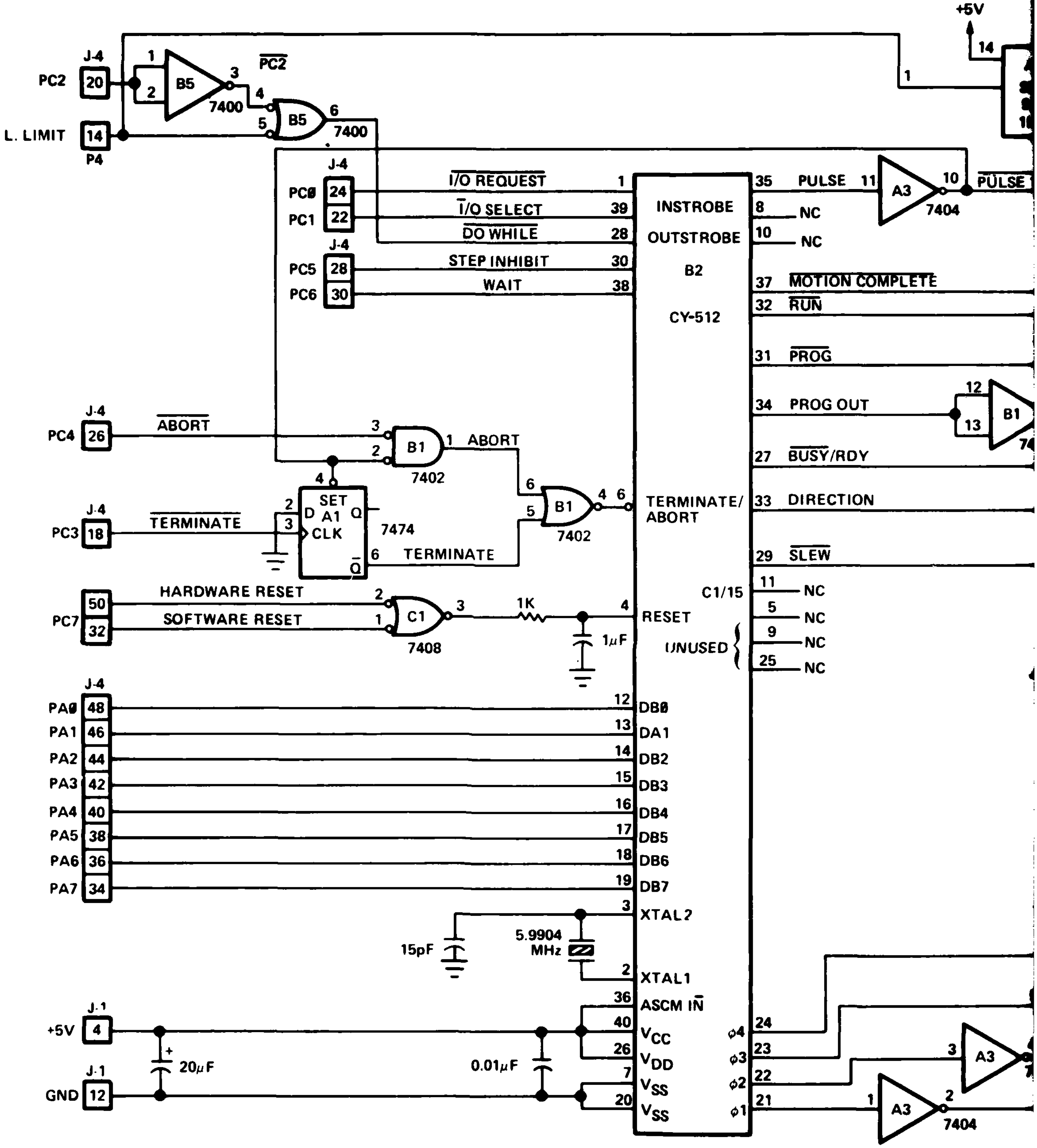


FIGURE III-11

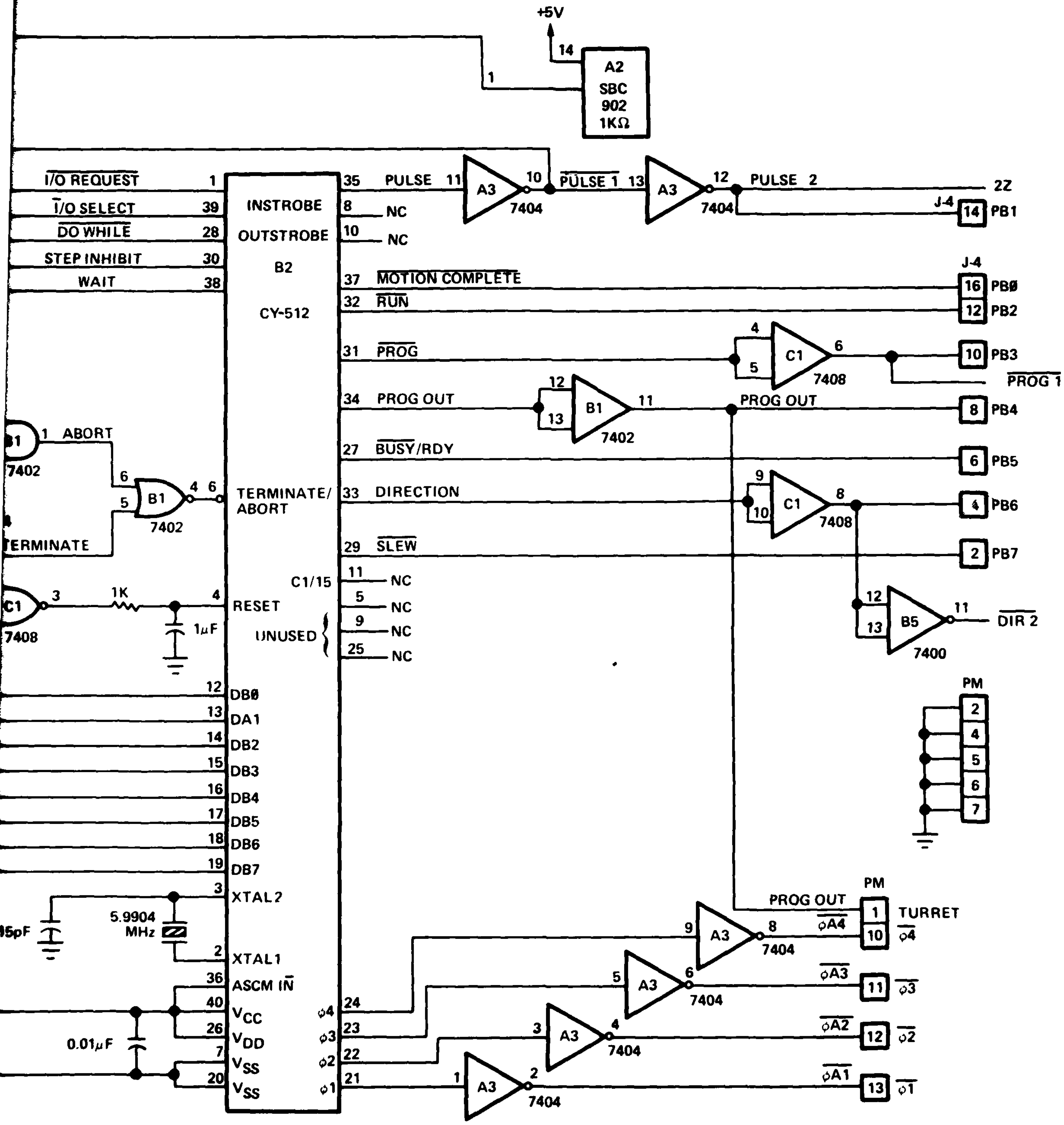
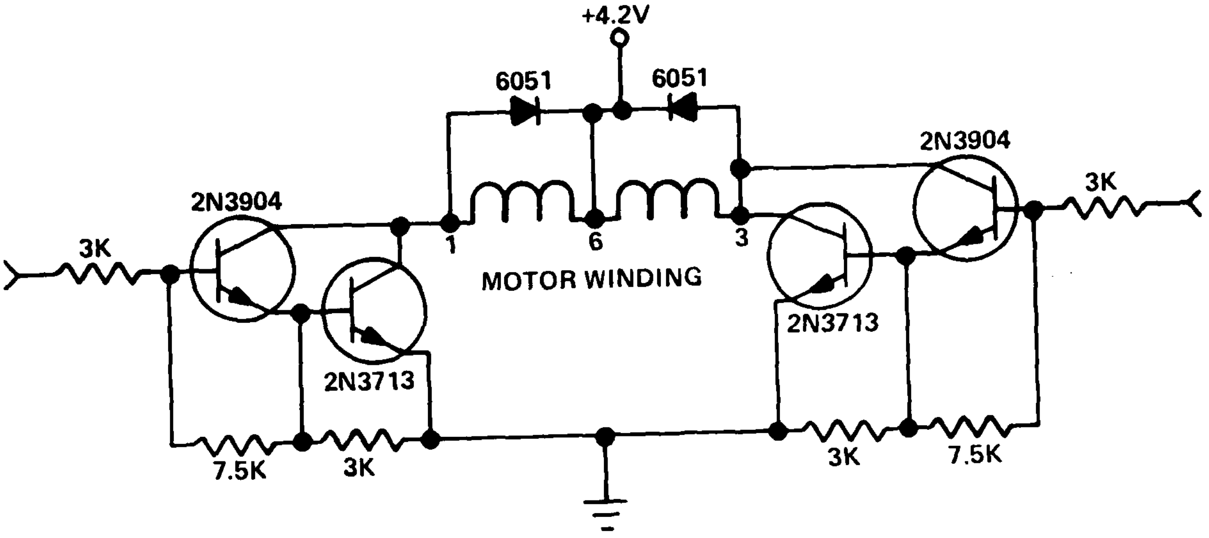


FIGURE III-11 STEPPER MOTOR CONTROLLER (CY-512 INTERFACE)

2



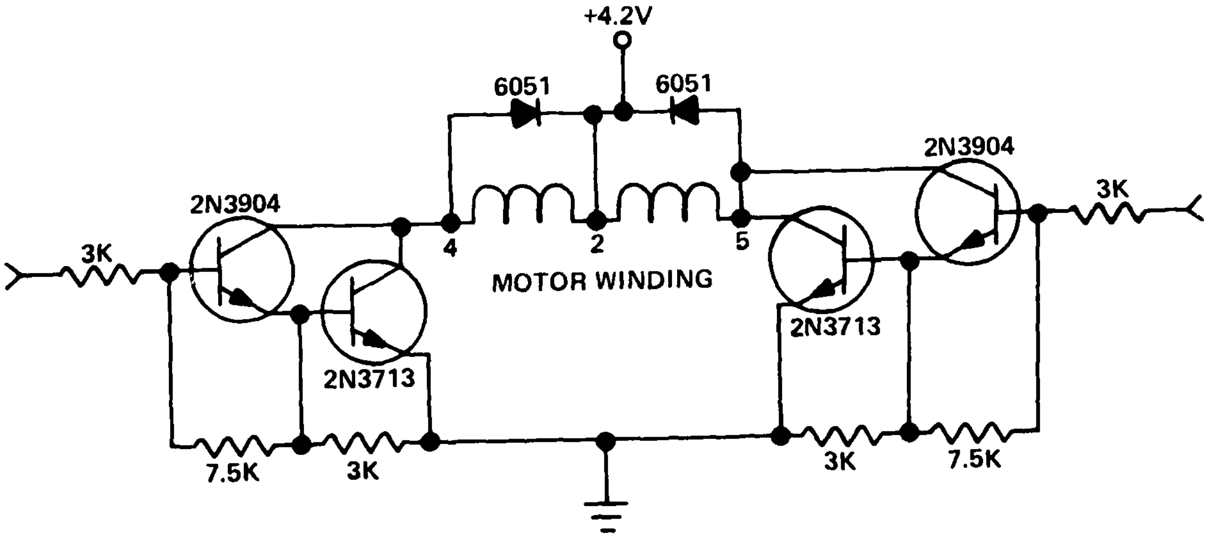


FIGURE III-12 STEPPER MOTOR DRIVERS

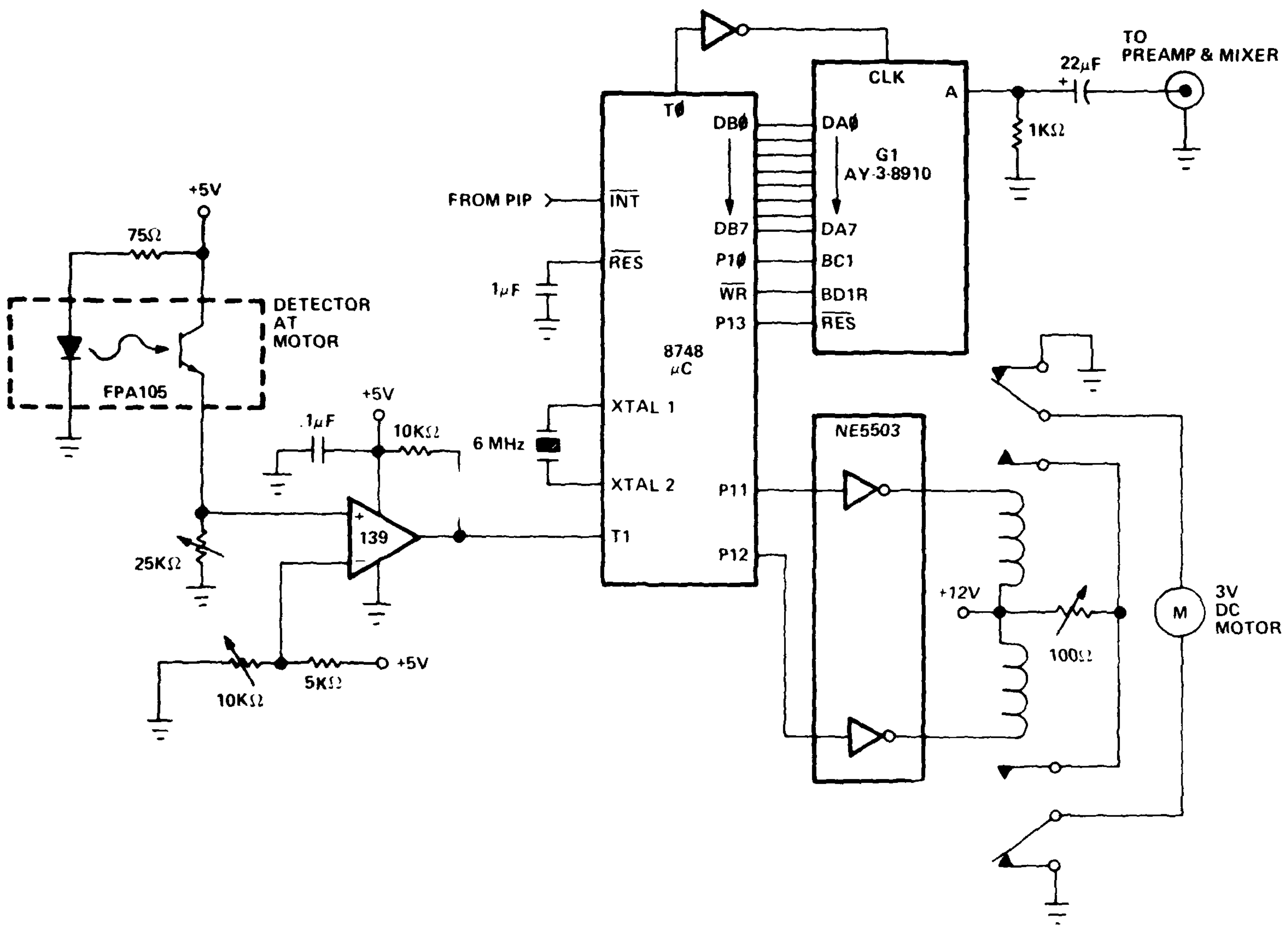


FIGURE III-13 TURRET CONTROLLER

A Superior Electric, Slo-Syn, Synchronous Stepping Motor with 200 steps per revolution was utilized.

The tank turret can be moved by using a software controlled pin on the CY-512. The CY-512 is fed into the circuit shown in Figure III-13. On software command the turret moves 90° toward the trainee and the tank gun fires. It then moves back to dead ahead after a short 2 or 3 second time delay.

F. Weight Loss and Recoil Mechanism

Launch effects of the DRAGON simulator are a very important facet of the training mission. Two of the launch transients which must be overcome by the DRAGON gunner are the weight loss due to the missile leaving the launch tube and the recoil of the launcher due to slight uncompensated differences in the pressures at launch. Weapon launch effects of weight loss and recoil are simulated via mechanical attachments to the DRAGON bipod.

The recoil mechanism is a sliding platten upon which the DRAGON bipod and gunner's feet are supported. The platten is covered with a rubber and steel hybrid material that allows the gunner to firmly plant the bipod legs in position and stabilize the launcher using his boots to press agains the bipod supports. At launch the platten is given an impulse from a pneumatic solenoid thus imparting a sensation of recoil to the launcher.

The weight loss simulation is accomplished by a weight mass that is attached to the bipod via a pivot and pneumatic cylinder. When the DRAGON simulator is armed for launch, the pneumatic cylinder is energized which in turn raises the weight and places an additional equivalent weight of the DRAGON missile on the shoulder of the DRAGON gunner through mechanical leverage. When the simulated missile is launched, the pneumatic cylinder is relaxed, thus releasing the weight and effectively removing the equivalent ROCKET weight from the gunner's shoulder. The recoil and weight loss circuit for driving the pneumatics is shown in Figure III-14.

G. Pull Down Measurement and Reticle Insertion for Gunner Sight Picture Display

Three LED indicator lights are present on the instructor's console to provide the instructor a quantitative indication of how much force a trainee places on the DRAGON tube and his shoulder. When the trainee produces a force on his shoulder equivalent to or greater than a predetermined threshold, the yellow LED comes on. If pull down force is increased beyond a second, higher threshold, a green LED comes on. If neither threshold is reached, a red LED is on. The threshold for the yellow and green LEDs are variable in a range of approximately 50 to 250 pounds force.

A circuit utilizing a strain gauge bridge was developed to generate a signal which is strictly the result of a force at the trainee's shoulder. The strain gauges used are manufactured by Wm. T. Bean, Inc. They are general purpose foil gauges constructed of Constantan with a polymide backing. Two of the strain gauges are strategically located on the DRAGON tube so as to unbalance the bridge only if the trainee has his shoulder properly positioned and is applying a downward force on the sight. The bridge, when unbalanced, supplies a DC level to a two stage DC amplifier. The amplified DC level is the input to two comparators. One comparator activates the yellow light when its threshold is breached while the other controls the green light. The circuit is shown in Figure III-15.

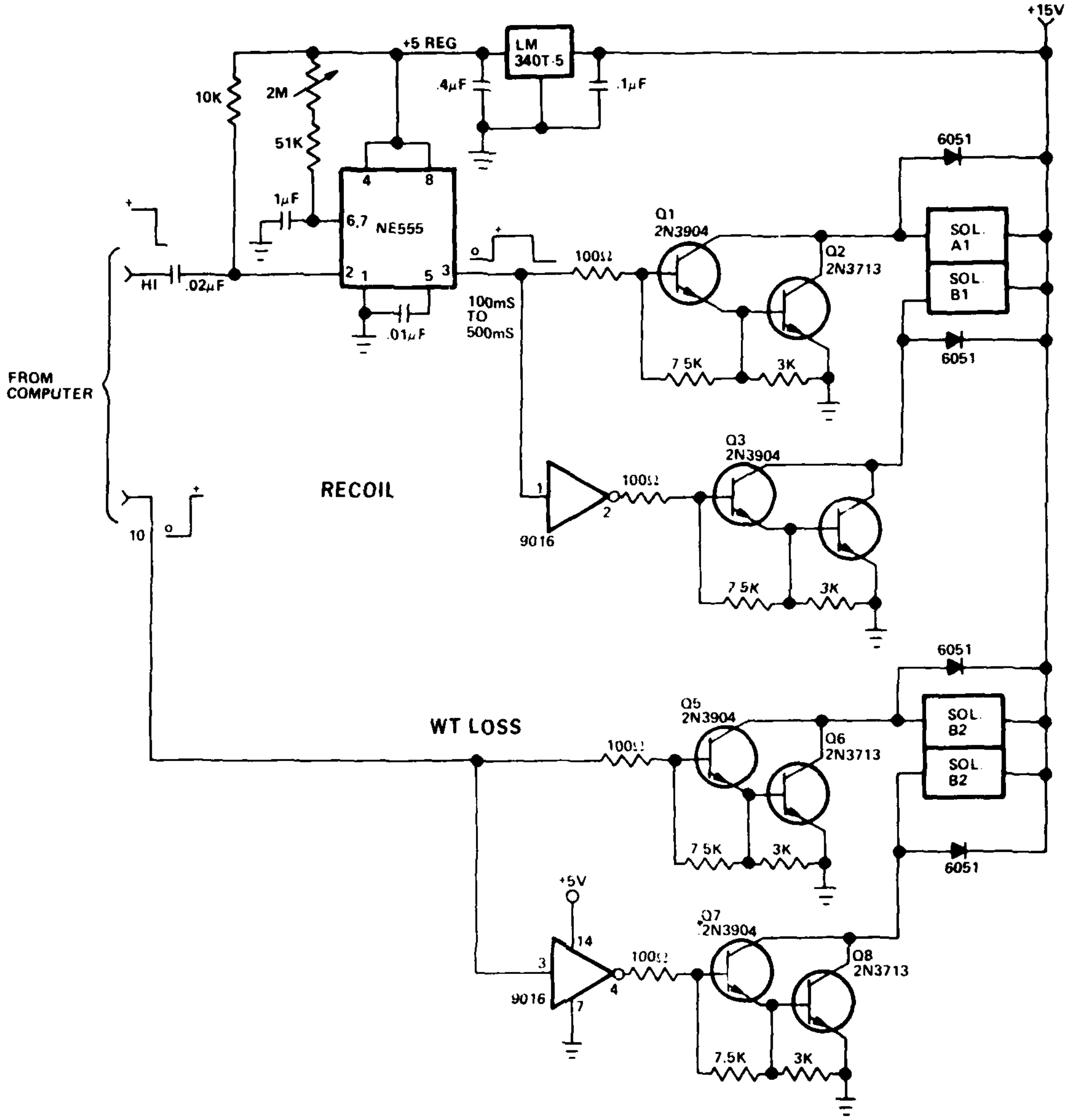


FIGURE III-14 RECOIL AND WEIGHT LOSS CIRCUIT

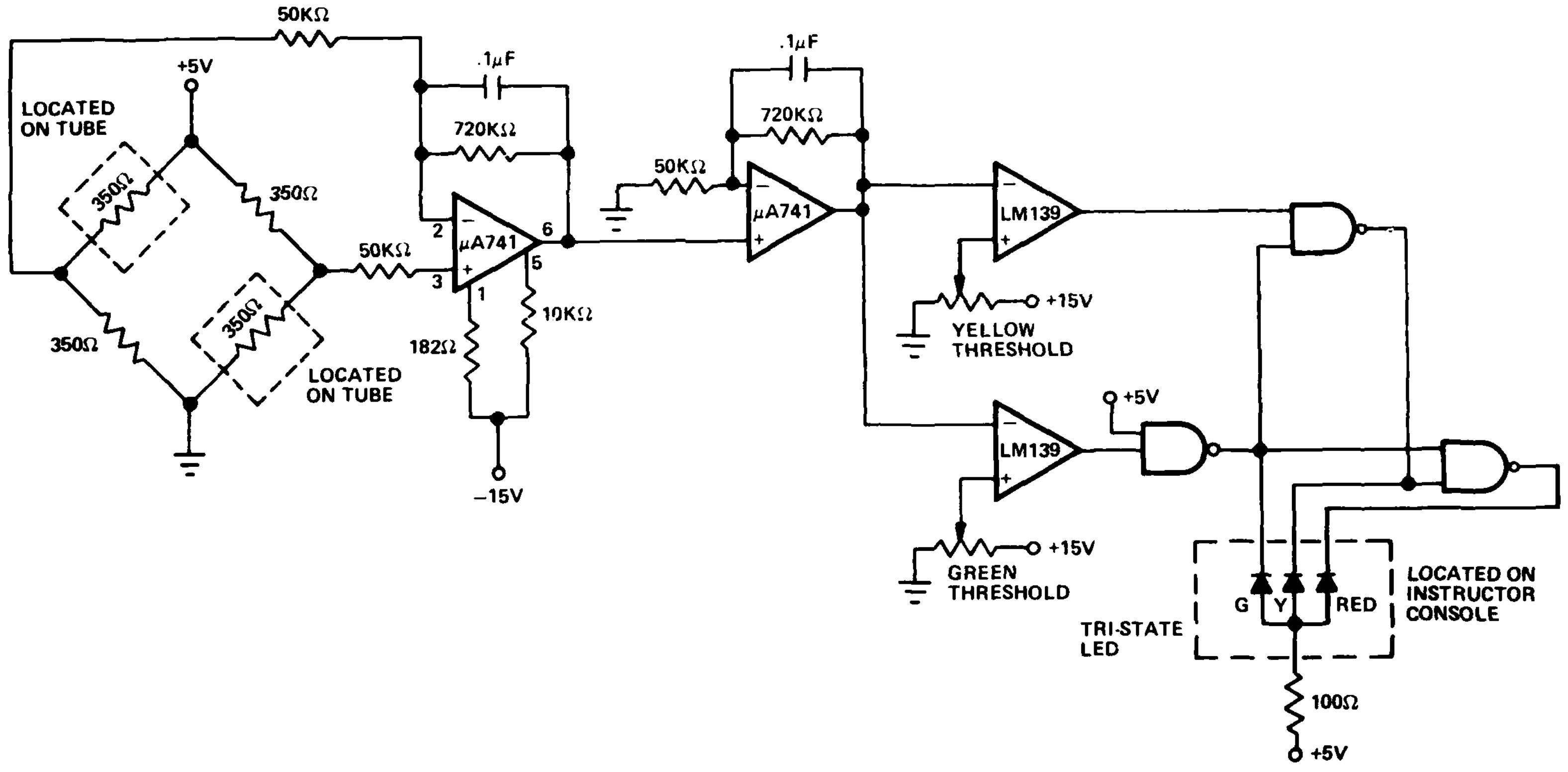


FIGURE III-15 PULL DOWN MEASUREMENT CIRCUIT

In order to provide more realism to the instructor's view of the gunner's sight picture, an electronic reticle was inserted into the instructor's TV monitor. A Cohu sync generator located inside the instructor's console provides drive signals to synchronize all the video signals throughout the system. The vertical and horizontal drive signals provide inputs to the reticle circuit. These signals each pass through a low pass active filter with a cut-off frequency centered at the repetition rate of the drive signal. In this way the square wave drive signals are filtered to provide sine wave outputs of frequency identical to the repetition rate of the inputs. The outputs of the filters input to voltage comparators which generate TTL square waves with falling edges adjustable about the midway times between two drive pulses. These falling edges trigger one shots which generates pulses whose duration determines the width of the retical lines. A horizontal reticle is produced by blanking out one or more lines of video. To insure that an entire line is blanked out and not a portion of it, a flip-flop further conditions the output of the horizontal line one shot. The clock for this flip-flop is provided by the vertical drive pulse which occurs for each line of video. An AND gate combines the output of the flip- flop (horizontal line) and the output of the vertical line one shot. This in turn controls an analog switch. The switch allows video to pass through to the TV, or when activated by the AND gate, sends a negative (black) signal to the TV. The position of the horizontal line is adjusted at the voltage comparator. The level of comparison, as it moves in relation to the sine wave input, controls the position of the falling edge at the output. The position of the vertical line is controlled by a phase shifter at the input of the voltage comparator. The reticle insertion circuit is shown in Figure III-16.

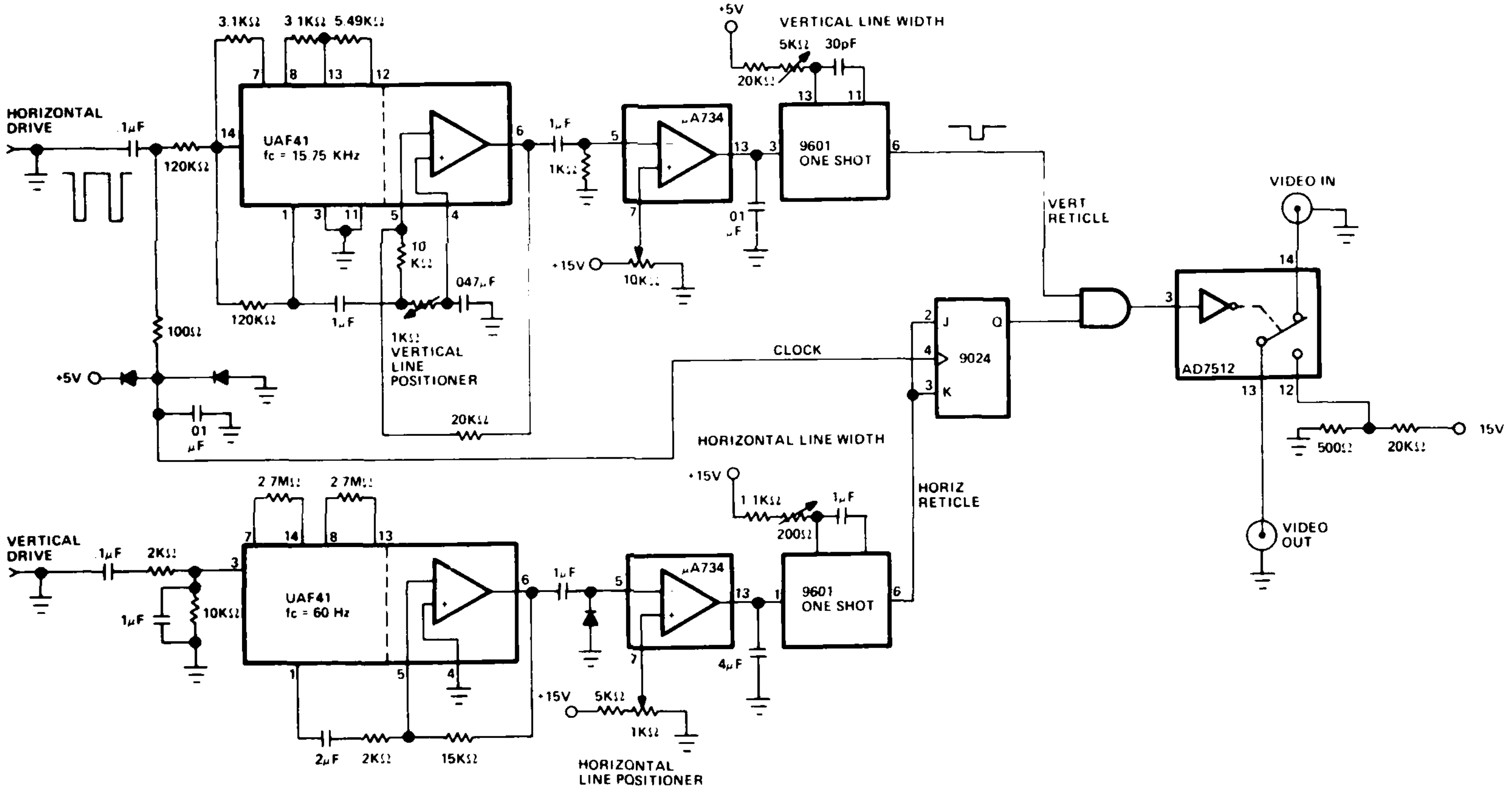


FIGURE III-16 RETICLE INSERTION CIRCUITRY

SECTION IV

CONCLUSIONS

This system has undergone preliminary evaluation by both Army and US Marine Corps experienced DRAGON gunners. All gunners were favorably impressed with its realism and teaching attributes.

The results of testing this device will be included in the final report.

Work is currently under way to include a TOW training capability as part of the STAGS system.

Development has also begun to add a speech synthesizer chip. This chip has a 274 word vocabulary and will be used to coach the student using verbal output.

Work has also started to simulate a thermal sight capability. This work will be covered in the final report.

APPENDIX A

DRAGON FLIGHT SIMULATOR EQUATIONS

The following data was provided by McDonnel Douglas Corporation under Contract N61339-80-M-3518 for use in simulating the DRAGON.

OBJECTIVE

Develop a set of simplified equations that will approximate the Dragon missile in flight when acted upon by the guidance commands as influenced by the gunner's aiming errors.

STUDY APPROACH

The equations currently programmed in the Dragon six degree of freedom simulator were examined and simplified as much as possible while still maintaining a statistically accurate representation of weapon performance. Some of the simplifying assumptions are:

(1) Missile dynamics based on a point mass solution

(2) Small angle approximation are used in the calculation of missile dynamics

(3) Effect of tracker sampling on missile trajectory while in the tracker linear field of view is negligible and is not simulated.

The six degree of freedom equations thus modified were exercised and compared to results from the six degree of freedom simulator. Modification to the thrust level and guidance parameters were made to tailor the trajectory to the more exact six degree of freedom results. Sufficient comparative analysis was conducted to assure that the simplified equations gave acceptable results over a range of crossing and stationary target conditions and with a variety of gunner aiming errors.

PROGRAM DESCRIPTION

At the beginning of each simulated flight, initial missile velocity and position is established in each of 3 orthogonal axes. The reference axes are established by the initial launch line. The target is placed on the launch line with a constant crossing velocity and time is set to zero.

Equations of motion are then solved every 0.02 seconds in each axis using gravity, drag and side thruster accelerations as inputs (see Figure A-1). At the end of each time increment, the new missile position (X, Y, Z) along with gunner aiming error (G1 and G2) and target position (E3) are seen as an angular input (E1, E2) into the tracker (See Figure A-2). The tracker calculates a side thruster firing angle based on the azimuth error (E1) and a firing frequency based on the elevation error (E2) and the commanded firing angle (W1).

When the tracker commands a firing, the resulting velocity change is resolved into each axis. This in turn changes missile position in a direction to reduce the sensed angular displacements between tracker and missile, thereby closing the guidance loop.

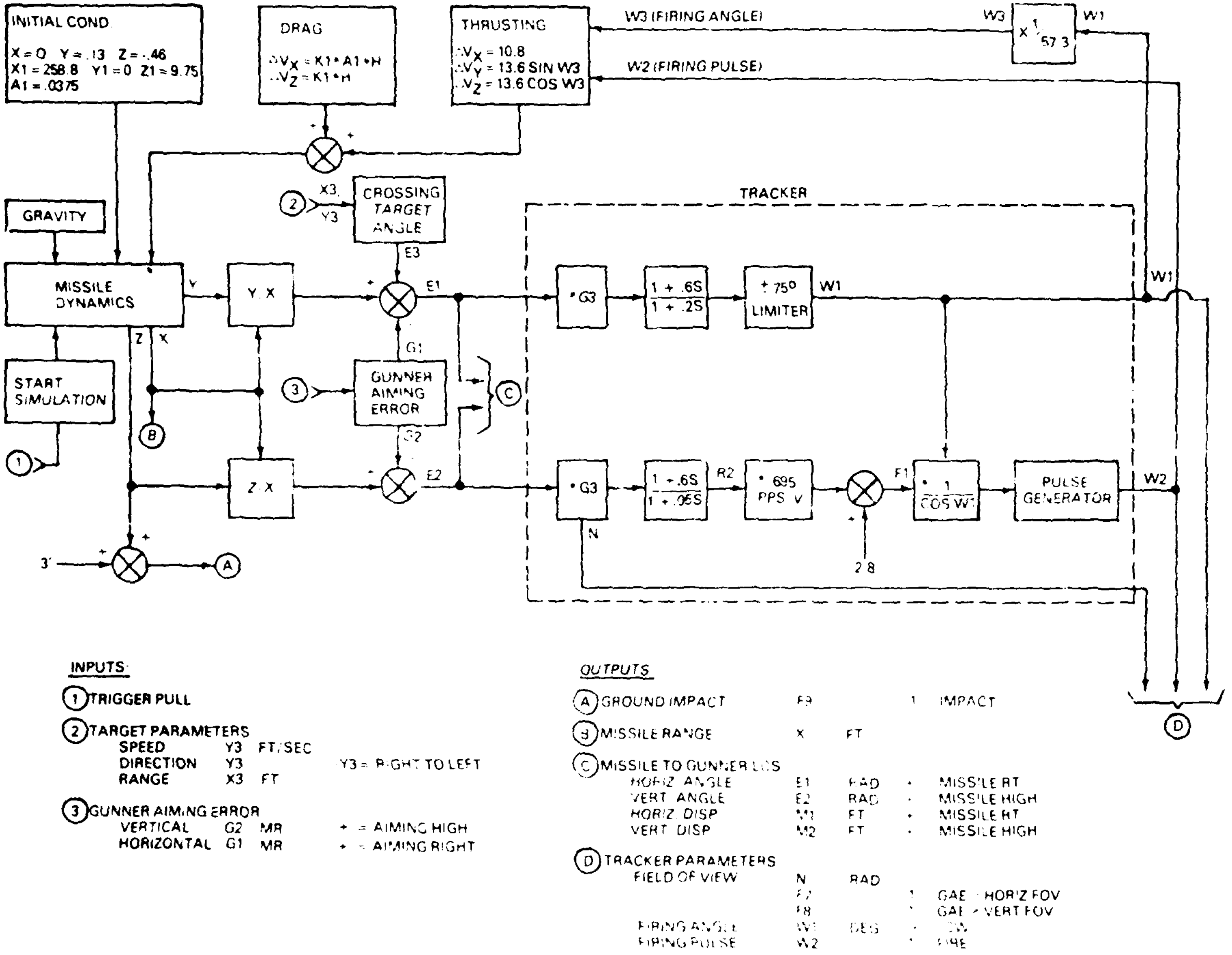
The program listing in BASIC language along with a symbol reference list and symbol descriptions are included.

DISCUSSION AND CONCLUSIONS

The equations presented herein will provide a good approximation of the flight characteristics of Dragon when subjected to the gunner aiming errors used in this analysis.

To interface this Dragon flight model with STAGS, several input parameters from the STAGS are necessary. The input parameters are trigger pull; target speed, direction and range; and gunner aiming error. Trigger pull will be initiated by the gunner using the Dragon training mockup, electrically sensed and transmitted to the Dragon flight simulator. This signal will be used to begin the simulated missile launch and subsequent flight. The target speed, direction, and range information are predetermined values for a given scenario. Target information is used in the Dragon flight simulator to generate system lag characteristics, i.e., the average distance at which the missile flies behind the target center. The STAGS will measure gunner aiming error with respect to a selected aim point on the tank target. This aiming error data is needed in the Dragon simulator to generate guidance error commands.

The sequence of events in a simulated flight begins with actuation of the tracker trigger. Figure A-3 is a time line or sequence of events for the launch phase based on nominal conditions at 70°F. This is an abbreviated time line which includes only those events which are relevant to the weapons trainer. Between trigger pull and gas generator ignition batteries are building up voltage and safe and arm functions are taking place; however, gas generator ignition at 515 milliseconds after trigger pull is important as an indicator of when to apply the simulation of sound pressure level. First motion of the missile in the launch tube begins at 533 milliseconds and could be used to initiate a simulation of weight shift ending at a tube exit time 20 milliseconds later. At tube exit, launch gas (fire and smoke) visible to the gunner is released. From this point the missile is flying ballistically and the missile flare is building up to its operating level. This flare is operating at 735 milliseconds allowing the tracker to recognize missile position relative to its line of sight and guidance corrections are generated in the tracker. At 1003 millisecond the warhead is armed providing an indication of what type of simulation of ground impact should be considered. Prior to this time a ground impact appears as a dud but beyond this time ground impact would cause the warhead to detonate and simulating this situation would be



INPUTS:

1 TRIGGER PULL

2 TARGET PARAMETERS

SPEED	Y3	FT/SEC	
DIRECTION	Y3		Y3 = RIGHT TO LEFT
RANGE	X3	FT	

3 GUNNER AIMING ERROR

VERTICAL	G2	MR	+ = AIMING HIGH
HORIZONTAL	G1	MR	+ = AIMING RIGHT

OUTPUTS

A GROUND IMPACT	F9		1	IMPACT
B MISSILE RANGE	X	FT		
C MISSILE TO GUNNER LOS				
HORIZ ANGLE	E1	RAD	+	MISSILE RT
VERT ANGLE	E2	RAD	+	MISSILE HIGH
HORIZ DISP	M1	FT	+	MISSILE RT
VERT DISP	M2	FT	+	MISSILE HIGH
D TRACKER PARAMETERS				
FIELD OF VIEW	N	RAD		
	F7		1	GAE > HOR'Z FOV
	F8		1	GAE > VERT FOV
FIRING ANGLE	W1	DEG	+	DOWN
FIRING PULSE	W2		+	FIRE

FIGURE A-1 SIMULATION BLOCK DIAGRAM

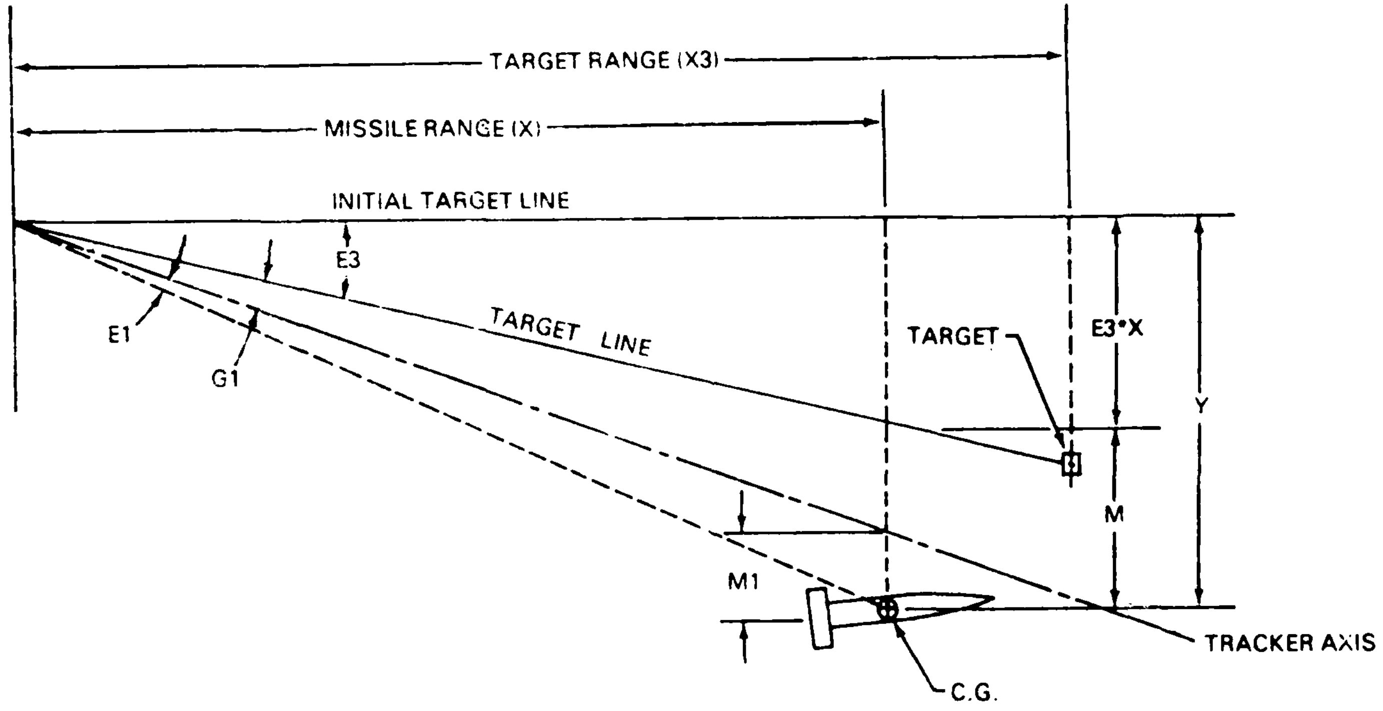


FIGURE A-2 HORIZONTAL PLANE GEOMETRY

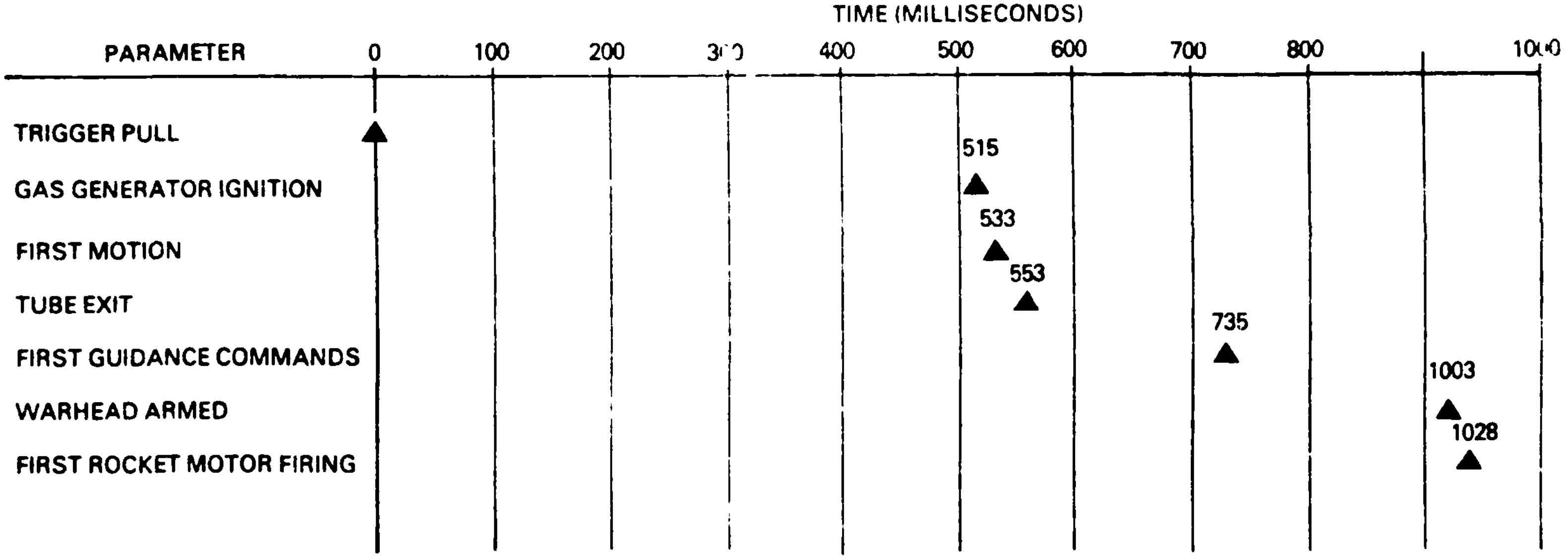


FIGURE A-3   DRAGON LAUNCH SEQUENCE TIME LINE quite different. The ballistic phase ends with the first rocket motor firing which is commanded automatically at a fixed time of 475 milliseconds from missile first motion. Subsequent to this first firing, the normal closed loop guidance is active in both the horizontal and vertical channels. The Dragon system was designed with a set of limits established for the aiming errors and the trainer may want to incorporate these limits in its scheme of scoring the gunners performance. Figure A-4 is a plot of these limits in both channels for stationary and crossing targets. Another factor that may be considered in modeling the Dragon flight is the influence of the trackers field of view; i.e., missiles' flare image relative to the center of the trackers' IR detector (coincident with gunners line of sight). As this flare image approaches the field of view boundaries guidance performance is degraded and when these boundaries are exceeded guidance is lost and the system goes into a squelch mode. In this mode of operation the tracker commands a constant firing angle and firing rate based on the last guidance information. A more accurate model of tracker performance which includes the field of view effect may be added at a later time but does require significantly more computing than this initial model.

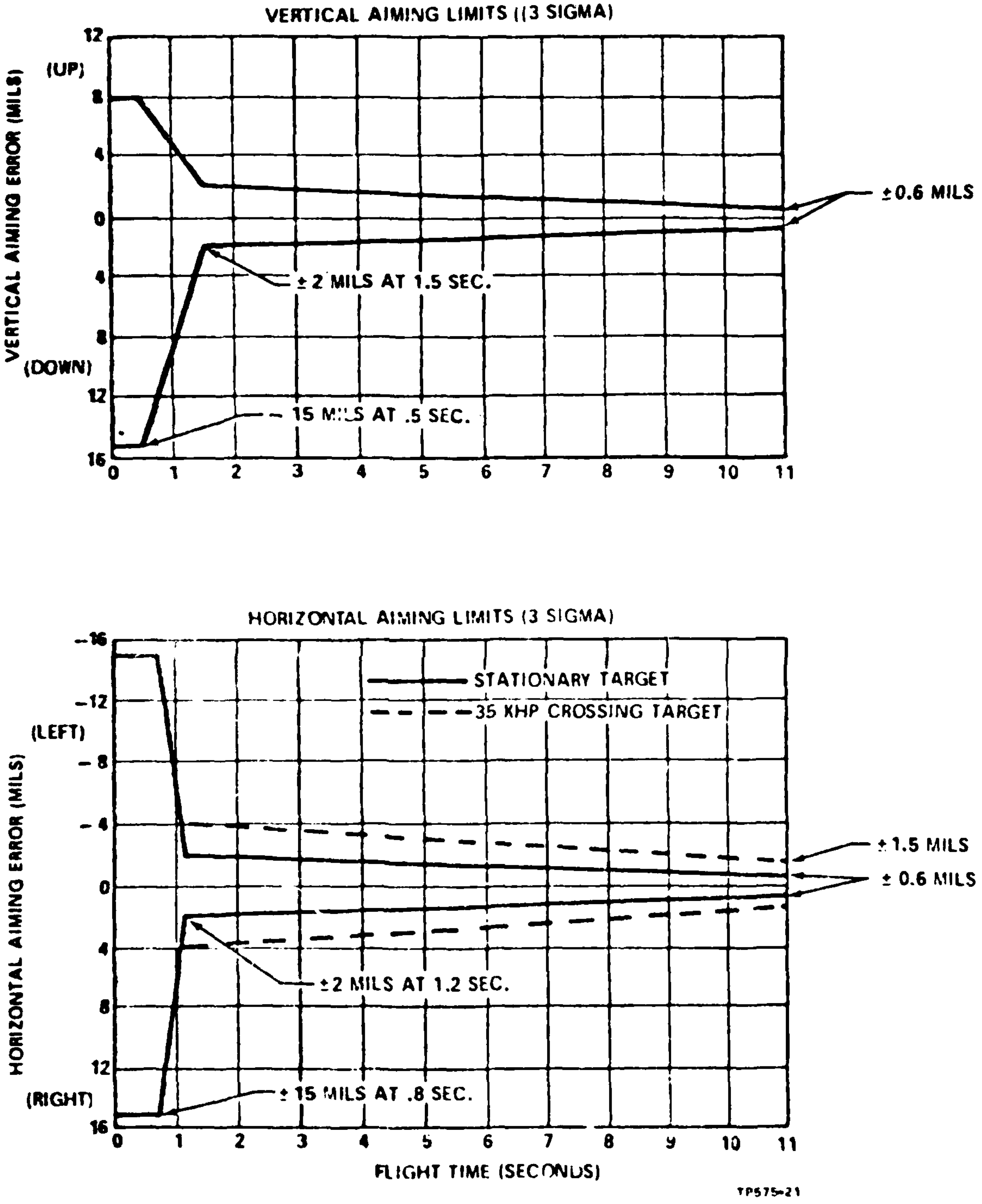


FIGURE A-4   GUNNER ERROR LIMITS

APPENDIX B

MULTI-PROCESSOR MAIN PROGRAMS

```
ISIS-II PL/M-86 V2 1 COMPILATION OF MODULE DRAGONFLIGHTMODULE
OBJECT MODULE PLACED IN  F1 DRAGFL OBJ
COMPILER INVOKED BY   PLM86  F1 DRAGFL 011 DEBUG ROM IXREF DATE (08/16/10)


   1          DRAGON$FLIGHT$MODULE  DO.

              /**************************************************************************
                                    OFF-BOARD ABSOLUTE ADDRESSES
              **************************************************************************/

   2   1      DECLAFF (YANG_BIRD, ZANG_BIRD) INTEGER AT (0A010H), (BIRD_DT_RDY,
                      FIRE_BIRD, BIRD_HITS, BIRD_MISSES, H_REP_RQ, H_REP_GO, V_REP_RQ,
                      V_REP_GO,GRND_BIRD,END_REPRISE) BYTE AT (0A014H);

              /********************** END OFF-BOARD ADDRESSES ***************************/

       1      DECLARE VELOCITY LITERALLY  34 .    /* TARGET VELOCITY
                         IN UNITS OF 0 1 MILLI-RADIANS PER SEC */

   4   1      DECLARE (GOING,ACTION) BYTE, PORT_B LITERALLY '0CAH',
                      DEL_BIG_H INTEGEF  BIG_H INTEGER.

   5   1      DECLARE (TCOUNT0,HTARG1,DEL_HTARG,DEL_HMISS,DEL_Y) INTEGER,TCNT WORD AT (0F000H);

   6   1      TARGET Y  PROCEDURE INTEGER.
   7   2        DECLARE (TCOUNT,TTCOUNT,HTCOUNT,TENS_COUNT) INTEGER.
   8   2        TCOUNT = INT ( NOT TCNT) - TCOUNT0.
   9   2        TENS COUNT = TCOUNT/10.
  10   2        HTCOUNT = TCOUNT/100.
  11   2        TTCOUNT = TCOUNT/1000.
  12   2        RETURN (TTCOUNT*267 + (HTCOUNT - TTCOUNT*10)*27 +
                          (TENS_COUNT - HTCOUNT*10)*3);
              /* MODEL MOVES 36 7/8" IN 5240 COUNTS   IT IS 22 FEET FROM TRAINEE
                 IT THEREFORE MOVES (36 875/22*12)*10000/5240 = 0 267 (0 1 MRAD/COUNT) */
  13   2        END TARGET_Y.

  14   1      TIME_DELAY  PROCEDURE (HOW_LONG) EXTERNAL;
  15   2       DECLARE HOW_LONG WORD.
  16   2       END TIME_DELAY.

  17   1      DECLARE RESULTS(800) STRUCTURE(S_COUNT INTEGER,S_X INTEGER,
                      S_Y INTEGER S_Z INTEGER, S_GAEY_I INTEGER, S_GAEZ_I INTEGER,
                      S_S$TCNT INTEGER, S_VMISS INTEGER, S_VOUT INTEGER) AT (2000H),
                      I INTEGER.

  18   1      DECLARE (B_Y,B_Z,DATA_RDY1) BYTE EXTERNAL;
  19   1      DECLARE SAVE_TCNT INTEGEF.

  20   1      H_REPRISE  PROCEDURE PUBLIC.
  21   2       SAVE_TCNT = 0.
  22   2       H_REF_RQ = 1.
  23   2       DO WHILE NOT H_REP_GO.
  24   3        END.
  25   2      H_REP_GO = 0.
  26   2       DO I = 0 TO COUNT - 1.
```

```
 27   3         IF RESULTS(I).S_S$TCNT > SAVE_TCNT THEN
 28   3          THRUSTER: DO;
 29   4           FIRE_BIRD = 1;
 30   4           SAVE_TCNT = RESULTS(I).S_S$TCNT;
 31   4          END THRUSTER;
 32   3         CALL TIME_DELAY (19);
 33   3         B_Y = LOW ( UNSIGN (100 - RESULTS(I).S_GREY_I));
 34   3         B_Z = LOW ( UNSIGN (RESULTS(I).S_Y + 100));
 35   3         DATA_RDY1 = 1;
 36   3         BIRD_DT_RDY = 1;
 37   3          END;
 38   2          END_REPRISE = 1;
 39   2          END H_REPRISE;

 40   1       V_REPRISE: PROCEDURE PUBLIC;
 41   2        SAVE_TCNT = 0;
 42   2        V_REP_RQ = 1;
 43   2        DO WHILE NOT V_REP_GO;
 44   3         END;
 45   2       V_REP_GO = 0;
 46   2        DO I = 0 TO COUNT - 1;
 47   3         IF RESULTS(I).S_S$TCNT > SAVE_TCNT THEN
 48   3          THRUSTER: DO;
 49   4           FIRE_BIRD = 1;
 50   4           SAVE_TCNT = RESULTS(I).S_S$TCNT;
 51   4          END THRUSTER;
 52   3         CALL TIME_DELAY (19);
 53   3         B_Y = LOW ( UNSIGN (100 - RESULTS(I).S_GAEZ_I));
 54   3         B_Z = LOW (UNSIGN (RESULTS(I).S_Z + 100));
 55   3         DATA_RDY1 = 1;
 56   3         BIRD_DT_RDY = 1;
 57   3          END;
 58   2          END_REPRISE = 1;
 59   2        END V_REPRISE;

 60   1       DECLARE GROUND_DUD  LITERALLY '6',
                      GROUND_EXP  LITERALLY '5',
                      HIT_TARGET  LITERALLY '3',
                      ROCKET_POP  LITERALLY '2';

 61   1       SOUND: PROCEDURE (WHAT_KIND) EXTERNAL;
 62   2        DECLARE WHAT_KIND BYTE;
 63   2        END SOUND;

              /* THIS VERSION OF "DRAGON" HAS BEEN TESTED AGAINST AN NTEC BASIC-80
              PROGRAM.  THE BASIC-80 PROGRAM, IN TURN, WAS CHECKED WITH THE MCDONNELL DOUGLAS
              VERSION OF TYPICAL DRAGON MISSILE FLIGHTS, WITH AND WITHOUT GUNNER ERRORS.
              NO SIGNIFICANT DIFFERENCES BETWEEN THE VARIOUS PROGRAMS HAVE BEEN FOUND */

              /* THE PROGRAM USES INTEGER MATH EXCLUSIVELY AND REQUIRES  2.12  MILLI-SEC TO
              EXECUTE EACH PASS THROUGH THE "FLIGHT" LOOP AS DETERMINED USING ICE86.  A SOFT-
              WARE, FLOATING-POINT-MATH VERSION WAS TRIED BUT REQUIRED ABOUT 2 SECONDS PER
              PASS.  THE PROGRAM MUST OPERATE IN REAL TIME TO SIMULATE THE MISSLE PERFORMANCE.
              TO DO THIS EACH PASS MUST REQUIRE NO MORE THAN 20 M-SECONDS.  THEREFORE, THE
              FLOATING-POINT VERSION WAS ABANDONED.

                 THE PURPOSE OF THE PROGRAM IS TO PROVIDE DRAGON CROSS-TRACK POSITION DATA
```

```
             TO THE SMALL "TV" VIEWER IN THE TRAINEE'S SIGHTING DEVICE  THE TV DRIVER
             WILL BE AN A SEPARATE PC BOARD PLUGGED INTO THE MULTIBUS   THIS WILL REQUIRE
             MULTI-PROCESSOR OPERATION AND ALL M P  PROTOCAL INCLUDING "MUTUAL EXCLUSION"
             MUST BE OBSERVED   THE REQUIREMENT, THEREFORE, REMAINS TO ESTABLISH A BLOCK
             OF MEMORY THAT CAN BE ACCESSED BY MULTIPLE MASTERS ACROSS THE MULTIBUS AND
             A  SEMAPHORE FLAG MUST BE PROVIDED FOR DATA PROTECTION. */

  64   1     DECLARE (GAEY, GAEZ) WORD EXTERNAL, (GAEY_I,GAEZ_I) INTEGER;

  65   1     DECLARE (X,Y,Z,VX,VY,VZ,GAMMA,PHI,DEL$VX,DEL$VY,DEL$VZ) INTEGER;

  66   1     DECLARE (H$FOV,V$FOV,FIRE) BYTE,
                     (GAIN$FAC,NUTAT,GAIN,REF$VOLT,COUNT) INTEGER;

  67   1     DECLARE (HOUT,INT$H,ERR$H,IN$HCOMP) INTEGER;

  68   1     DECLARE (VOUT,INT$V,ERR$V,IN$VCOMP) INTEGER;

  69   1     DECLARE (HANG,VANG,HMISS,VMISS) INTEGER;

  70   1     DECLARE (HTARG,VTARG,XTARG) INTEGER;

  71   1     DECLARE FINISHED BYTE PUBLIC, FIRST$FIRE BYTE,
                     (S$TCNT,INT$FF,COS$PHI,F$FREQ) INTEGER;

  72   1     DECLARE (H_MISS_HEX,V_MISS_HEX,X_MISS_HEX) INTEGER PUBLIC,
                     (IDEAL_S$TCNT,TARGET_S$TCNT) INTEGER PUBLIC;

  73   1     DECLARE (DEL_OFF_H,OFF_H) INTEGER;

  74   1     DECLARE SMOOTH(4) INTEGER, II BYTE;

                     /* S$TCNT IS THE NUMBER OF SIDE THRUSTER FIRINGS
                        COS$PHI: UNIT = 0.001
                        F$FREQ : UNIT = 0.001 FIRE/SEC
                        INT$FF : UNIT = 0.001 FIRINGS
                        FINISHED IS THE FLAG FOR A HIT GROUND
                        COUNT : UNIT = 0.02 SECOND
                        GAIN$FAC : UNIT = 0.01 X MCDONNELL BASIC PROGRAM "G" UNIT
                        NUTATE : UNIT = 10^-4 RADIAN
                        GAIN : UNIT = 0.1 VOLT / RADIAN
                        REF$VOLT : UNIT = 0.01 VOLT   */

                     /* H$OUT : UNIT = 0.002 VOLT -- BASIC PROGRAM R1 VALUE
                        INT$H : UNIT = 0.002 VOLT -- BASIC PROGRAM 01 VALUE
                        GAEY_I : UNIT = 1/2 PIXEL (IS .25 MRAD) -- BASIC PROG  G1(RAD)
                              LENS FOCAL LENGTH = 125 MM --- SEE DRAGON VAR
                        ERR$H : UNIT = 0.002 VOLT -- BASIC PROGRAM I1 VALUE
                        IN$HCOMP : UNIT = 0.002 VOLT BASIC PROGRAM P1 VALUE
                        V$OUT : UNIT = 0.002 VOLT -- BASIC PROGRAM R2 VALUE
                        INT$V : UNIT = 0.002 VOLT -- BASIC PROGRAM 02 VALUE
                        GAEZ_I : UNIT = 1/2 PIXEL (IS .25 MRAD) -- BASIC PROG. G2(RAD)
                        ERR$V : UNIT = 0.002 VOLT -- BASIC PROGRAM I2 VALUE
                        IN$VCOMP : UNIT = 0.002 VOLT -- BASIC PROGRAM P2 VALUE
                        HMISS & VMISS = 0.1 MRAD -- G1 + E1 & G2 + E2 ANGLE FROM
                                        MISSILE TO TARGET LINE
                        XTARG  UNIT = 2 INCHES -- NOT IN BASIC
                     HTARG & VTARG: UNIT = 0 1 MRAD  -- BASIC E1 & E2


                 * HANG : UNIT 0 1 MILLIRADIAN --- BASIC "E1"
                  VANG : UNIT 0.1 MILLIRADIAN --- BASIC "E2"
                     BIG_H ; UNIT 0 1 MIL-RAD/50
                  VANG_BIRD: UNIT 0 1 MILLIRADIAN = HANG
                  HANG_BIRD: UNIT 0 1 MILLIRADIAN = VANG */

            /* X IN 2 INCHES. VX IN 2 INCHES PER SECOND, Y & Z IN UNITS OF 0 05 INCHES.
              VY & VZ IN UNITS OF 0 20 INCHES PER 10 SECONDS  GAMMA & PHI ARE IN DEGREES */

            /* OFF_H IS THE HORIZONTAL DISTANCE OFF TARGET LINE  UNITS OF 0.05 INCHES */

            /* THIS PROGRAM ASSUMES THAT EVERY CYCLE REQUIRES 0.02 SECONDS  IT MUST
            BE CORRECTED WHEN WE DETERMINE HOW LONG THE THING TAKES */

            /* ************************************************************
                         PROGRAM VARIABLE INITIALIZATION
              ********************************************************** */
  75   1    INITIATE$VAR: PROCEDURE PUBLIC;

  76   2       TCOUNT0 = INT(NOT TCNT);     /* TCNT COMES INVERTED FROM CY512 BRD */
  77   2       HTARG1 = 0;      /* HTARG FROM PREVIOUS PASS */
  78   2       X = 0;
  79   2       OFF_H, HMISS = 0;
  80   2       SMOOTH(0),SMOOTH(1),SMOOTH(2),SMOOTH(3) = 0;
  81   2       Z = -110;
  82   2       VX = 1559;
  83   2       VY = 0;
  84   2       VZ = 5850;
  85   2       GAMMA = 2;
  86   2       PHI,HOUT,INT$H,GREY_I,VOUT,INT$V,GREZ_I,HANG,VANG,S$TCNT,INT$FF = 0;
  87   2       HTARG = TARGET_Y;
  88   2       VTARG = 0
  89   2       XTARG = 19685;        /* THIS WILL CHANGE WITH FILM DATA NOW = 1000 M */

  90   2       IDEAL_S$TCNT = XTARG/720;  /***** ASSUMES FIXED XTARG **********/

  91   2       FIRE,FINISHED,FIRST$FIRE,BIRD_HITS,BIRD_MISSES,GRND_BIRD = 0;  /* NOTE- TYPE IS ALL BYTE HERE */
  92   2       H_REP_R0,H_REF_GO,V_REP_R0,V_REP_GO,END_REPRISE = 0;
  93   2       DEL_HMISS = 0;
  94   2       I = 0;
  95   2       COUNT = 0;
  96   2     END INITIATE$VAR;

            /*********************** PROGRAM STARTS ******************************/

  97   1    FLIGHT: PROCEDURE PUBLIC;

             ***********************************************************************
                                    MISSLE DATA SAMPLE
            ********************************************************************* */

  98   2    H_MISS_HEX = (SMOOTH(0)+SMOOTH(1)+SMOOTH(2)+SMOOTH(3))/4;

  99   2    RESULTS(I).S_COUNT=COUNT;
```

```
RESULTS(I).S_X=0;
RESULTS(I).S_Y = H_MISS_HEX;
IF RESULTS(I).S_Y < -95 THEN RESULTS(I).S_Y = -95;
IF RESULTS(I).S_Y > 150 THEN RESULTS(I).S_Y = 150;
RESULTS(I).S_Z = ((X/100)*VMISS)/50;          /* DITTO */
IF RESULTS(I).S_Z < -95 THEN RESULTS(I).S_Z =  -95;
IF RESULTS(I).S_Z > 150 THEN RESULTS(I).S_Z =  150;
RESULTS(I).S_GAEY_I=GAEY_I
RESULTS(I).S_GAEZ_I=GAEZ_I;
RESULTS(I).S_S$TCNT=S$TCNT;

DO II = 0 TO 2;
  SMOOTH(3 - II) = SMOOTH(2 - II);
  END;
I=I+1

/*********************************************************************
                             TINY TANK POSITION
*********************************************************************/


HTARG = TARGET_Y;
DEL_HTARG = HTARG - HTARG1;
HTARG1 = HTARG;           /* SAVE TARGET POSITION TILL NEXT PASS */

COUNT = COUNT + 1;

/* *********************************************************************
                             MISSILE  DYNAMICS
**********************************************************************/

ZANG_BIRD = VANG;         /* TRANSFER BIRD POSITIONS TO "B_BOARD " */
YANG_BIRD = HANG;
BIRD_DT_RDY = 1;

DEL$VZ = -3*GAMMA - 386;

IF FIRE = 1 THEN DO;
  DEL$VZ = DEL$VZ + 7715 - PHI*PHI;
  CALL SOUND(ROCKET_POP);
  FIRE_BIRD = 1;
  END;

GAEY_I = SIGNED(GAEY);
GAEZ_I = SIGNED(GAEZ);

TARGET_S$TCNT = S$TCNT;
X_MISS_HEX = X-XTARG;
IF IABS(X_MISS_HEX) < 25 THEN   /* 20 ==> 3 & 1/3 FT FROM RANGE OF TARGET */
 AT_TARGET:  DO;
  IF IDEAL_S$TCNT > TARGET_S$TCNT THEN IDEAL_S$TCNT = TARGET_S$TCNT;
  IF X < 2000H THEN
       V_MISS_HEX = ((X/100)*VMISS)/600;  /* VERT MISS DISTANCE, FT  */
     ELSE V_MISS_HEX = ((X/1000)*VMISS)/60;
  IF  IABS(H_MISS_HEX) < 48 AND IABS(V_MISS_HEX) < 3 THEN  /* IN 8X6 FT RECTANGLE */
     GOOD_SHOT:  DO;
       CALL SOUND(HIT_TARGET);
```

```
146   4              BIRD_HITS = 1;
147   4              BIRD_MISSES = 0;
148   4              FINISHED = 1
149   4              END GOOD_SHOT;
150   3            ELSE BAD_SHOT: BIRD_MISSES = 1;
151   3         END AT_TARGET;

152   2       VZ = VZ + DEL$VZ / 2;
153   2       Z = Z + VZ / 125;
154   2       VZ = VZ + DEL$VZ / 2;

155   2       DEL$VX = -2;
156   2       IF FIRE = 1 THEN DEL$VX = DEL$VX + 62;
158   2       VX = VX + DEL$VX / 2;
159   2       DEL_X = VX/50;
160   2       X = X + DEL_X;
161   2       VX = VX + DEL$VX / 2;

162   2       GAMMA = VZ / ( VX*2 );

163   2       IF FIRE = 1 THEN DEL$VY = -((21598-PHI*PHI)/160)*PHI;
165   2          ELSE DEL$VY = 0;
166   2       VY = VY + DEL$VY / 2;
167   2       DEL_OFF_H = VY/125 - (DEL_X/10)*(HMISS/25) - (X/250)*DEL_HTARG;
168   2       OFF_H = OFF_H + DEL_OFF_H;
169   2       SMOOTH(0) = OFF_H/20;
170   2       VY = VY + DEL$VY / 2;
171   2       FIRE = 0;

              /******************************************************************************
                                            DEFINING ANGLES
              ******************************************************************************/

              /* WE INTRODUCE THE TARGET DATA FOR TARGET MOVEMENT FROM I.C. */

172   2       IF COUNT > 5 THEN  /* TO AVOID A "DIVIDE BY ZERO" WHEN CALCULATING "HMISS" */
173   2       ANGLES: DO;
174   3         HMISS = (2*OFF_H)/(X/120);
175   3         VMISS = (2*Z)/(X/128) - VTARG;
176   3         HANG = HMISS - 5*GAEY_I/2;
177   3         VANG = VMISS - 5*GAEZ_I/2;
178   3          END ANGLES;


              /******************************************************************************
                                     THIS IS THE TRACKER GAIN SECTION
              ******************************************************************************/

179   2       GAIN$FAC = (1446 - 2 * COUNT)/10 ;
180   2       IF GAIN$FAC > 140 THEN GAIN$FAC = 140;
182   2       IF GAIN$FAC < 100 THEN GAIN$FAC = 100;
184   2       IF COUNT < 94 THEN NUTAT = 240;      /* WITH 125MM LENS, FOV IS 24 MRAD */
186   2           ELSE NUTAT = ((28595/GAIN$FAC)*100)/COUNT;
187   2       IF COUNT > 47 THEN REF$VOLT = 875;
189   2           ELSE REF$VOLT = 18 * COUNT;
190   2       IF REF$VOLT > 875 THEN PEF$VOLT = 875;
192   2       IF COUNT < 46 THEN GAIN = 43 * COUNT;
```

```
194   2           ELSE GAIN = 110 * ( 8750 / NUTAT );

             /*****************************************************************
                               CHECK FOR NUTATION RADIUS EXCEEDED
             *****************************************************************/
195   2      H$FOV,V$FOV = 0;
196   2      IF COUNT > 20 THEN
197   2      CHECK$FOV: DO;
198   3           IF IABS(HANG) > NUTAT THEN H$FOV = 1;
200   3           IF IABS(VANG) > NUTAT THEN V$FOV = 1;
202   3      END CHECK$FOV;

             /*****************************************************************
                          THIS IS THE HORIZONTAL RATE POSITION PORTION
             *****************************************************************/

203   2      IF NOT H$FOV THEN
204   2      CALC$H: DO;
205   3           IF COUNT < 22 THEN IN$HCOMP = ((GAIN/10)*HANG)/20; /* COUNT=22 <==> TIME= 44 S
                                                         & FIRST FIRE OCCURS @  474 S */
207   3                ELSE IN$HCOMP = (GAIN/200)*HANG;
208   3              END CALC$H;
209   2        ELSE IN$HCOMP = (REF$VOLT/5)*HANG/IABS(HANG);

210   2      ERR$H = IN$HCOMP - HOUT;
211   2      HOUT = INT$H/10 + 3*IN$HCOMP;
212   2      INT$H = INT$H + ERR$H;

             /*******************************************************************
                                        HORIZONTAL WIRE
             *******************************************************************/

213   2      PHI = HOUT/50;
214   2      IF PHI < -75 THEN PHI = -75;
216   2      IF PHI > 75 THEN PHI = 75;

             /*******************************************************************
                            THIS IS THE VERTICAL RATE POSITION PORTION
             *******************************************************************/

218   2      IF NOT V$FOV THEN
219   2      CALC$V: DO;
220   3           IF COUNT < 22 THEN IN$VCOMP = ((GAIN/10)*VANG)/20;
222   3                ELSE IN$VCOMP = (GAIN/200)*VANG;
223   3              END CALC$V;
224   2        ELSE IN$VCOMP = (REF$VOLT/5)*VANG/IABS(VANG);

225   2      ERR$V = IN$VCOMP - VOUT;
226   2      VOUT = INT$V/5 + 9*IN$VCOMP;
227   2      INT$V = INT$V + ERR$V;

228   2      ERR$V = IN$VCOMP - VOUT;   /* HERE WE TAKE INTEGRATION @ 1/2 OF "COUNT" TWICE */
229   2      VOUT = INT$V/5 + 9*IN$VCOMP;
230   2      INT$V = INT$V + ERR$V;

             /*****************************************************************
                                     FIRE PULSE GENERATION
```

```
                 ***************************************************************************/

 231   2        F$FREQ = (14000 - 7*VOUT)/5.
 232   2        COS$PHI = 995 - (PHI*PHI)/7.

 233   2        IF COUNT = 24 THEN
 234   2        FIRST$SHOT: DO.
 235   3          FIRE = 1.
 236   3          S$TCNT = 1.
 237   3          INT$FF = 0.
 238   3          FIRST$FIRE = 1.
 239   3          END FIRST$SHOT.

 240   2        IF FIRST$FIRE THEN
 241   2        CANFIRE: DO.
 242   3             IF S$TCNT < 30 THEN
 243   3        SOME$LEFT: DO.
 244   4                IF INT$FF > COS$PHI THEN
 245   4        FIRE$ONE: DO.
 246   5                   INT$FF = 0
 247   5                   FIRE = 1.
 248   5               S$TCNT = S$TCNT +1.
 249   5               END FIRE$ONE.
 250   4                INT$FF = INT$FF + F$FREQ/50.
 251   4                END SOME$LEFT
 252   3            END CANFIRE.

 253   2        IF Z < -720 THEN
 254   2        GROUNDED: DO.
 255   3          FINISHED(GRND_BIRD) = 1.
 256   3          IF COUNT > 23 THEN CALL SOUND (GROUND_EXP):
 258              ELSE CALL SOUND (GROUND_DUD).
 259   3            END GROUNDED.

 260   2        END FLIGHT.

 261   1        END DRAGON$FLIGHT$MODULE.


MODULE INFORMATION

     CODE AREA SIZE     = 0AD3H   2771D
     CONSTANT AREA SIZE = 0000H      0D
     VARIABLE AREA SIZE = 0082H    130D
     MAXIMUM STACK SIZE = 0008H      8D
     430 LINES READ
     0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION
```

```
ISIS-II MCS-86 MACRO ASSEMBLER V2 1 ASSEMBLY OF MODULE IR_CENTER
OBJECT MODULE PLACED IN  F1 ORAGIR OBJ
ASSEMBLER INVOKED BY   ASM86  F1 ORAGIR 001 DEBUG DATE (02/24/09)


LOC  OBJ                  LINE     SOURCE

                             1     NAME    IR_CENTER
                             2
                             3     ;THIS IS A DRIVER PROGRAM FOR THE RETICON 6020 USING THE 86/12 BOARD
                             4     ;IT IS BEING UPDATED 9/23/81
                             5
                             6     DGROUP  GROUP   DATA, STACK, SBC_REGS, XFER_SEG
                             7     CGROUP  GROUP   CODE
                             8
                             9     ASSUME  SS DGROUP, CS CGROUP, DS DGROUP, ES DGROUP
                            10
----                        11     STACK   SEGMENT STACK   'STACK'
0000 (64                    12             DW      64 DUP(?)
     ????
     )
0080                        13     TOP_STK LABEL   WORD
----                        14     STACK   ENDS
                            15
                            16     ;THIS PROGRAM WILL RESIDE ON THE SBC "DFS"  IT WILL WRITE Y, Z(CENTER)
                            17     ;DATA TO THE SBC "PIP" VIA THE MULTIBUS  THE "PIP" HAS BEEN "JUMPERED"
                            18     ;SO AS TO ALLOW THE MULTIBUS TO ACCESS 8K OF ITS RAM STARTING AT LOCATION
                            19     ;A000H. THE "DFS" JUMPERS ALLOW MULTIBUS ACCESS TO 8K OF RAM STARTING
                            20     ;AT 8000H  THE ON-BOARD LOCATION OF THESE AVAILABLE 8K-S START AT 6000H
                            21     ;ON BOTH BOARDS. THE BOARDS (REF FIG 2-1 86/12) ARE JUMPERED AS FOLLOWS:
                            22     ;
                            23     ;  SBC DFS JUMPERS MULTIBUS ACCESS    SBC PIP JUMPERS MULTIBUS ACCESS
                            24     ;    -------------------------         --------------------------
                            25     ;       127-128 ==> X = 0                    127-128 ==> X = 0
                            26     ;    S1 6-11 CLOSED                       S1 6-11 CLOSED
                            27     ;    S1 5-12   "    ==> 8K                S1 5-12   "    ==> 8K
                            28     ;    S1 1-16   "                          S1 1-16   "
                            29     ;    S1 2-15 OPEN                         S1 2-15 OPEN
                            30     ;    S1 3-14 CLOSED                       S1 3-14 CLOSED
                            31     ;    S1 4-13   "    ==> 8000H             S1 4-13 OPEN  ==> A000H
                            32     ;
                            33
----                        34     XFER_SEG        SEGMENT AT 0A00H        ;ON SBC "PIP" AS NOTED ABOVE
                            35                                            ;WILL PASS DATA FOR YCNTR,ZCNTR
                            36     ;IF THE "DFS" IS THE DATA "SUPPLIER" AND THE "PIP" IS THE "USER "
                            37
0000 (1                     38     START_BIT       DB      1 DUP(?)
     ??
     )
                            39             PUBLIC  B_Y,B_Z,DATA_RDY1
0001 (1                     40     B_Y             DB      1 DUP(?)
     ??
     )
0002 (1                     41     B_Z             DB      1 DUP(?)
     ??
     )
0003 (1                     42     DATA_RDY1       DB      1 DUP(?)
LOC  OBJ                  LINE     SOURCE

      ]
0004 01                     43     BAD_MISS       DB       1 DUP(?)
      ??
      ]
0005 01                     44     OFFSET_Y       DB       1 DUP(?)
      ??
      ]
0006 01                     45     OFFSET_Z       DB       1 DUP(?)
      ??

----                        46     XFER_SEG       ENDS
                            47
----                        48     SBC_REGS       SEGMENT COMMON
0000 320                    49     SBCREG  DB     800 DUP(?)
      ??
      ]
0320 01                     50     MISS    DB     1 DUP(?)                 THE FOLLOWING 6 BYTES ARE NAMED "PARTLY_OFF"
                                    IN RT86XF
      ??

                            51
0321 01                     52     RIGHT   DB     1 DUP(?)
      ??
      ]
0322 01                     53     LEFT    DB     1 DUP(?)
      ??
      ]
0323 01                     54     UP      DB     1 DUP(?)
      ??
      ]
0324 01                     55     DOWN    DB     1 DUP(?)
      ??
      ]
0325 01                     56             DB     1 DUP(?)        ;DUMMY BYTE TO MAKE 3 WORDS OF "PARTLY_OFF" IN RT86XF
      ??
      ]
0326 01                     57     YCNTR   DB     1 DUP(?)        ;THE FOLLOWING 6 BYTES ARE IN "LOCATIONS" IN RT86XF
      ??
      ]
0327 01                     58     ZCNTR   DB     1 DUP(?)
      ??
      ]
0328 01                     59     YMAX    DB     1 DUP(?)
      ??
      ]
0329 01                     60     YMIN    DB     1 DUP(?)
      ??
      ]
032A 01                     61     ZMAX    DB     1 DUP(?)
      ??
      ]
032B 01                     62     ZMIN    DB     1 DUP(?)
      ??
      ]
```

```
LOC OBJ                  LINE     SOURCE

                          63
                          64                PUBLIC   GREY,GREZ
[illegible]               65      GREY      DW       1 DUP(?)

[illegible]               66      GREZ      DW       1 DUP(?)

                          67
                          68      PUBLIC   GREY,GREZ
                          69
----                      70      SBC_REGS           ENDS
                          71
                          72
                          73      EXTRN    RD_RAST   NEAR
 0064                     74      SIZ       EQU      64H
----                      75      CODE      SEGMENT PUBLIC  'CODE'
                          76
                          77                PUBLIC   SETRET
0000                      78      SETRET    PROC     NEAR
                          79                EXTRN    INIT1    NEAR
0000 55                   80                PUSH     BP
0001 1E                   81                PUSH     DS
0002 B8----         R     82                MOV      AX,DGROUP
0005 8ED8                 83                MOV      DS,AX
0007 8EC0                 84                MOV      ES,AX
0009 E80000         E     85                CALL     INIT1
000C A00000         R     86      WAIT0:    MOV      AL,START_BIT  ;THIS WAIT LOOP HOLDS UP RETICON DATA UNTIL
                          87                                       ; THE RETRO GRAPHICS BOARD IS SET UP AND THE
                          88                                       ; MATROX BOARD IS READY AND WAITING
000F 3C01                 89                CMP      AL,1
0011 75F9                 90                JNE      WAIT0
0013 1F                   91                POP      DS
0014 5D                   92                POP      BP
0015 C3                   93                RET
                          94      SETRET    ENDP
                          95
                          96                PUBLIC   VZCNTR
0016                      97      VZCNTR    PROC     NEAR
0016 55                   98                PUSH     BP
0017 1E                   99                PUSH     DS
0018 B8----         R    100                MOV      AX,DGROUP
001B 8ED8                101                MOV      DS,AX
001D 8EC0                102                MOV      ES,AX
001F E80000         E    103                CALL     RD_RAST
                         104
                         105      ;WE NOW LOOK FOR THOSE TRANSITIONS WHICH CAN BE ASSOCIATED WITH A SINGLE
                         106      ;BRIGHT SPOT ON THE RETICON CAMERA 100X100 FIELD OF VIEW. THE SPOT WILL BE
                         107      ;LOCATED AT THE CENTER OF THE TARGET. THE RETICON CAMERA WILL BE MOUNTED ON
                         108      ;THE CENTER LINE OF THE SIMULATED WEAPON SO THAT THE OFFSET OF THE SPOT FROM
                         109      ;THE CENTER OF THE 100X100 FIELD OF VIEW WILL MEASURE THE LEAD AND ELEVATION
                         110      ;OF THE WEAPON AT TRIGGER TIME. A SAMPLE RETICON DATA LINE CONTAINING A
                         111      ;SINGLE BRIGHT SPOT IS
                         112      ;
                         113      ;       07 07 1C 6A 1E EA 64 60
```

```
LOC  OBJ                  LINE     SOURCE

                          114
                          115      ;THIS INDICATES THREE TRANSITIONS IN THE LINE BUT ONLY THE FIRST TWO ARE
                          116      ;SIGNIFICANT. AT 1CH THERE IS A DARK-TO-LIGHT TRANSITION, AS INDICATED BY THE
                          117      ;HIGHEST ORDER BIT (HOB) = 0 IN DATA BYTE #3 (60). THE FOLLOWING TRANSITION
                          118      ;IS AT 1EH AND IS A LIGHT-TO-DARK, AS INDICATED BY HOB=1 IN THE FOLLOWING
                          119      ;BYTE #5 (E0). EVERY LINE HAS A FORCED TRANSITION AT THE END-OF-LINE,
                          120      ;LOCATION 64H=100D. THE PROGRAM STARTS FROM LINE #0 LOOKING FOR 03 AS THE
                          121      ;INITIAL DATA BYTE. IF 03 IS NOT FOUND, THE NEXT LINE IS EXAMINED. THIS IS
                          122      ;CONTINUED UNTIL 100 LINES HAVE BEEN EXAMINED FOR THE PROPER NUMBER OF
                          123      ;TRANSITIONS
                          124
                          125      ;RESISTER USAGE IN "CENTER"
                          126      ;       (AL) = NUMBER OF TRANSITIONS IN DATA LINE
                          127      ;       (AH) = BYTES TO ADD TO DATA LINE POINTER
                          128      ;       (BX) = DATA LINE POINTER. POINTS TO START OF DATA LINE IN "SBCREG"
                          129      ;       (CL) = LINE NUMBER
                          130      ;       (CH) = 64H = 100 ==> THE LAST DATA LINE
                          131
0022 B90165               132      CENTER  MOV    CX,6501H         ;(CH)=65H,1 + LAST LINE. (CL)=1:--> FIRST LINE
0025 BBFEFF               133              MOV    BX,-2            ;INITIAL VALUE OF DATA LINE POINTER
0028 B402                 134              MOV    AH,2             ;INITIAL DATA LINE POINTER INCREMENT
002A C606290364      R    135              MOV    YMIN,SIZ         ;SET INITIAL VALUE AT 64H
002F C6062B0364      R    136              MOV    ZMIN,SIZ         ;DITTO FOR ZMIN
0034 C606280301      R    137              MOV    YMAX,1
0039 C6062A0301      R    138              MOV    ZMAX,1
                          139
003E 3AE9                 140      DUMP    CMP    CH,CL            ;HAVE WE FINISHED WITH LAST LINE?
0040 7503                 141              JNE    OVER             ;NEED "OVER" BECAUSE CONDITIONAL JUMPS MUST BE
                          142                                       ;LESS THAN +127 BYTES AWAY
0042 E96E00               143              JMP    DONE
0045 02DC                 144      OVER    ADD    BL,AH            ;UPDATE DATA LINE POINTER. NOW BECAUSE WE CANNOT
0047 80D700               145              ADC    BH,0             ;ADD A SINGLE BYTE TO BX, WE DO IT IN TWO STEPS
                          146                                       ;USING THE CARRY FLAG, "CY". N.B. (BX)=0 ON
                          147                                       ;THE FIRST PASS THROUGH "DUMP."
                          148
004A 8A870000        R    149              MOV    AL,SBCREG[BX]    ;FIRST DATA BYTE ==> TRANSITIONS IN DATA LINE
                          150
004E 8AE0                 151              MOV    AH,AL            ;WILL FORM DATA LINE POINTER INCREMENT IN AH
0050 FEC4                 152              INC    AH
0052 D0E4                 153              SHL    AH,1             ;(AH)=2(AL+1), THE DATA LINE POINTER INCREMENT
                          154
0054 8A870200        R    155              MOV    AL,SBCREG[BX+2]
0058 3C64                 156              CMP    AL,SIZ
005A 7405                 157              JE     SKIP             ;IF NO SPOT, THEN GO TO NEXT DATA LINE
005C E80A00               158              CALL   GOODLN           ;WILL UPDATE SPOT INFORMATION
005F EBDD                 159              JMP    DUMP             ;GO TO NEXT DATA LINE
                          160
0061 3ACD                 161      SKIP    CMP    CL,CH            ;THE LAST LINE?
0063 7440                 162              JZ     DONE             ;YES! SO WE JUMP TO THE FINAL CLEAN-UP
0065 FEC1                 163              INC    CL               ;NO! SO WE RETURN TO "DUMP" AND
0067 EBD5                 164              JMP    DUMP             ;EXAMINE THE NEXT LINE
                          165
0069 80BF000002      R    166      GOODLN  CMP    SBCREG[BX],2     ;ONLY TWO TRANSITIONS?
006E 7419                 167              JE     TWOX             ;IF SO, SPOT IS ON RIGHT EDGE
0070 3A062907        R    168              CMP    AL,YMIN          ;IF NOT, GET NORMAL CENTER. RECALL THAT
```

```
LOC  OBJ                 LINE     SOURCE

                          169                                    ;AL CONTAINS SBCREG[BX+2]
0074 77                   170            JA       N1             ;JUMP IF (AL) IS ABOVE YMIN, IE  CY FLAG = 0
                          171
                          172     ;RECALL THAT A CMP OPERATION SUBTRACTS THE SOURCE OR 2ND OPERAND FROM THE
                          173     ;DESTINATION OR 1ST OPERAND  IT DOES THIS BY ADDING THE TWOS COMPLEMENT OF THE
                          174     ;SOURCE OPERAND TO THE DESTINATION OPERAND AND A CARRY-OUT FROM THE HIGH ORDER
                          175     ;BIT CAUSES THE CY FLAG TO SET TO 0, BECAUSE OF THE SUBTRACTION OPERATION
                          176     ;AT LEAST THIS IS WHAT THE 8080 DOES, AND THIS FLAG IS THE SAME  IN THE 8086
                          177     ;THEY SAY THAT AFTER A SUBTRACT OPERATION "CY" IS SET UPON A CARRY INTO(!)
                          178     ;THE HOB OF THE RESULT!
                          179
0076 A22903         R     180            MOV      YMIN,AL        ;IF NO JUMP THEN UPDATE VALUE OF YMIN
0079 8A870400       R     181     N1     MOV      AL,SBCREG[BX+4] ;RIGHT EDGE OF SPOT
007D 3A062803       R     182            CMP      AL,YMAX
0081 7218                 183            JB       N4             ;JUMP IF (AL) IS BELOW YMAX, OR CY = 1
0083 A22803         R     184            MOV      YMAX,AL        ;IF NO JUMP THEN UPDATE YMAX
0086 EB1390               185            JMP      N4             ;JUMP AROUND "TWOX"
0089 C606280364     R     186     TWOX   MOV      YMAX,64H       ;WILL BE ON RIGHT EDGE
008E 8A870200       R     187            MOV      AL,SBCREG[BX+2]
0092 3A062903       R     188            CMP      AL,YMIN
0096 7703                 189            JA       N4
0098 A22903         R     190            MOV      YMIN,AL
009B 3A0E2A03       R     191     N4     CMP      CL,ZMAX        ;AT THIS POINT Z IS MEASURED DOWNWARD
009F 7204                 192            JB       N3             ;AND WE MUST COMPLEMENT Z AT END
00A1 880E2A03       R     193            MOV      ZMAX,CL        ;UPDATE ZMAX
00A5 3A0E2B03       R     194     N3     CMP      CL,ZMIN
00A9 7704                 195            JA       N9
00AB 880E2B03       R     196            MOV      ZMIN,CL        ;UPDATE ZMIN
00AF FEC1                 197     N9     INC      CL
00B1 C3                   198            RET
                          199
00B2 C606200300     R     200     DONE   MOV      MISS,0
00B7 803E2B0364     R     201            CMP      ZMIN,64H
00BC 750D                 202            JNE      N5
00BE C606200301     R     203            MOV      MISS,1         ;ZMIN = 64H ==> NO SPOT, SO SHOT WAS A MISS!
00C3 C606040001     R     204            MOV      BAD_MISS,1     ;WE ARE "REALLY FINISHED"
00C8 EB7E90               205            JMP      READ
00CB 803E2B0301     R     206     N5     CMP      ZMIN,1
00D0 7505                 207            JNE      N6
00D2 C606230301     R     208            MOV      UP,1           ;SPOT INCLUDED FIRST LINE ==> SHOT WAS HIGH!
00D7 803E2A0364     R     209     N6     CMP      ZMAX,64H
00DC 7505                 210            JNE      N7
00DE C606240301     R     211            MOV      DOWN,1         ;SPOT INCLUDED LAST LINE ==> SHOT WAS LOW
00E3 803E290301     R     212     N7     CMP      YMIN,1
00E8 7505                 213            JNE      N8
00EA C606220301     R     214            MOV      LEFT,1
00EF 803E280364     R     215     N8     CMP      YMAX,64H
00F4 7505                 216            JNE      FINI
00F6 C606210301     R     217            MOV      RIGHT,1
                          218
00FB B065                 219     FINI   MOV      AL,65H
00FD 2A062A03       R     220            SUB      AL,ZMAX
0101 B465                 221            MOV      AH,65H
0103 2A262B03       R     222            SUB      AH,ZMIN
0107 A22B03         R     223            MOV      ZMIN,AL
```

```
LOC  OBJ                  LINE     SOURCE

010A 88262A03       R      224              MOV     ZMAX,AH
010E A02803         R      225              MOV     AL,YMAX
0111 02062903       R      226              ADD     AL,YMIN
0115 8A1E0500       R      227              MOV     BL,OFFSET_Y     ;HORIZONTAL BORESIGHT OFFSET
0119 02C3                  228              ADD     AL,BL
011B A22603         R      229              MOV     YCNTR,AL        ;YCNTR IN HALF-PIXELS FROM LEFT SIDE OF SCREEN
011E A20100         R      230              MOV     B_Y,AL
0121 B464                  231              MOV     AH,100
0123 2AE0                  232              SUB     AH,AL
0125 8AC4                  233              MOV     AL,AH
0127 98                    234              CBW
0128 A32C03         R      235              MOV     GREY,AX         ;GREY +IVE TO THE RIGHT  CAMERA INVERTS
012B A02A03         R      236              MOV     AL,ZMAX
012E 02062B03       R      237              ADD     AL,ZMIN
0132 8A1E0600       R      238              MOV     BL,OFFSET_Z
0136 02C3                  239              ADD     AL,BL
0138 A22703         R      240              MOV     ZCNTR,AL        ;ZCNTR IN HALF-PIXELS FROM BOTTOM OF SCREEN
013B A20200         R      241              MOV     B_Z,AL
013E B464                  242              MOV     AH,100
0140 2AE0                  243              SUB     AH,AL
0142 8AC4                  244              MOV     AL,AH
0144 98                    245              CBW
0145 A32E03         R      246              MOV     GREZ,AX         ;GREZ +IVE IF HIGH. REF. MCDONNEL-DOUGLAS
0148 B001                  247     READ     MOV     AL,1
014A A20300         R      248              MOV     DATA_RDY1,AL    ;THIS TELLS THE SLAVE PROCESSOR THAT NEW
                           249                                      ;DATA ARE READY.
014D 1F                    250              POP     DS
014E 5D                    251              POP     BP
014F C3                    252              RET
                           253
                           254     YZCNTR   ENDP
                           255
----                       256     CODE     ENDS
                           257
                           258              END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

```
ISIS-II PL/M-86 V2.1 COMPILATION OF MODULE MAIN_DRAGON_MODULE
OBJECT MODULE PLACED IN :F1:DRAGMN.OBJ
COMPILER INVOKED BY:  PLM86 :F1:DRAGMN.PLM DEBUG ROM IXREF DATE (10/01/10)


   1          MAIN_DRAGON_MODULE: DO;

              /***********************************************************************
                                         OFF-BOARD ADSOLUTE ADDRESSES
              ***********************************************************************/

   2   1      DECLARE (H_MIS_ASCII,V_MIS_ASCII,X_MIS_ASCII) (16) BYTE AT (0A020H),
                     FELL_SHORT BYTE AT (0A01EH);

              /******************** END OFF-BOARD ABSOLUTE ADDRESSES *****************/

   3   1      DECLARE (H_MISS_HEX,V_MISS_HEX,X_MISS_HEX) INTEGER EXTERNAL, I BYTE;

   4   1      SHOW_THRUSTERS: PROCEDURE EXTERNAL;
   5   2        END SHOW_THRUSTERS;

   6   1      MISS_COMMENT: PROCEDURE(HEX$ADR, DEC$ADR, DIRECTION) EXTERNAL;
   7   2           DECLARE (HEX$ADR, DEC$ADR) POINTER, DIRECTION BYTE;
   8   2       END MISS_COMMENT;

   9   1      DECLARE RT LITERALLY '0', LT LITERALLY '1',
                     UP LITERALLY '2', DN LITERALLY '3',
                     SH LITERALLY '4';

  10   1      DECLARE FOREVER LITERALLY 'WHILE 1', NO_TRIG_PUL LITERALLY
                  'INPUT(PORT_B) AND 1', ACTION BYTE;

  11   1      V_REPRISE: PROCEDURE EXTERNAL;
  12   2        END V_REPRISE;

  13   1      H_REPRISE: PROCEDURE EXTERNAL;
  14   2        END H_REPRISE;

  15   1      TIME_DELAY: PROCEDURE (HOW_LONG) EXTERNAL;
  16            DECLARE HOW_LONG WORD;
  17           END TIME_DELAY;

  18   1      SETRET: PROCEDURE EXTERNAL;
  19   2        END SETRET;

  20   1      SOUND: PROCEDURE(KIND) EXTERNAL;
  21   2        DECLARE KIND BYTE;
  22   2        END SOUND;

  23   1      DECLARE INITIAL_BANG LITERALLY '7', GYRO_START LITERALLY '1';

  24   1      DECLARE ARM LITERALLY '2', DISARM LITERALLY '3',
                     DROP LITERALLY '4', NO_DROP LITERALLY '5',
                     FINISHED BYTE EXTERNAL;

  25   1      DECLARE PORT_A LITERALLY '0C8H',PORT_B LITERALLY '0CAH';
```

```
                    PORT_C LITERALLY '0CCH', CONTROL LITERALLY '0CEH';

 26   1      YZCNTR: PROCEDURE EXTERNAL;
 27   2       END YZCNTR;

 28   1      PPI_SET: PROCEDURE EXTERNAL;    /* SETS UP PORTS_A & _C OUTPUT, PORT_B INPUT */
 29   2       END PPI_SET;

 30   1      FLIGHT: PROCEDURE EXTERNAL;
 31   2       END FLIGHT;

 32   1      INITIATE$VAP: PROCEDURE EXTERNAL;
 33   2       END INITIATE$VAP;

             /**************************************************************
                                   PROGRAM STARTS
             **************************************************************/

 34   1      START_UP:
               CALL PPI_SET;

 35   1      OUTPUT(CONTROL) = NO_DROP   /* DRIVES PORT_C BIT-2 HIGH & J1-20 LOW */

 36   1      OUTPUT(CONTROL) = ARM;  /* DRIVES PORT_C BIT-1 LOW & "J1"-22 HIGH */
                 /* PROBABLY NOT NEEDED --- PORT_C COMES UP LOW FOLLOWING PPI_SET */
 37   1      DO I = 0 TO 95;
 38   2        H_MIS_ASCII(I) = ' ';
 39   2      END;

 40   1      CALL INITIATE$VAP;       /* FOR DRAGON FLIGHT */

 41   1      DO WHILE NO_TRIG_PUL;
 42   2        END;

 43   1      CALL SOUND(GYRO_START);

 44   1      CALL TIME_DELAY (560);

 45   1      OUTPUT(CONTROL) = DROP;

 46   1      OUTPUT(CONTROL) = DISARM;

 47   1      CALL SOUND(INITIAL_BANG);

 48   1      CALL SETPET;

 49   1      DRAGON_FLYS: DO WHILE NOT FINISHED;
 50   2        CALL YZCNTR;
 51   2        CALL FLIGHT;
 52   2      END DRAGON_FLYS;

 53   1      IF H_MISS_HEX >  2 THEN CALL MISS_COMMENT(@H_MISS_HEX,@H_MIS_ASCII,RT);
 55   1      IF H_MISS_HEX < -2 THEN CALL MISS_COMMENT(@H_MISS_HEX,@H_MIS_ASCII,LT);
 57   1      IF V_MISS_HEX >  2 THEN CALL MISS_COMMENT(@V_MISS_HEX,@V_MIS_ASCII,UP);
 59   1      IF V_MISS_HEX < -2 THEN CALL MISS_COMMENT(@V_MISS_HEX,@V_MIS_ASCII,DN);

 61   1      IF Z_MISS_HEX < -24 THEN
 62   1        DROP_SHORT: DO;
 63   2         X_MISS_HEX = IABS(X_MISS_HEX)/6;
 64   2         CALL MISS_COMMENT(@X_MISS_HEX,@X_MIS_ASCII,SH);
 65   2         FELL_SHORT = 1;
 66   2        END DROP_SHORT;
 67   1      ELSE FELL_SHORT = 0

 68   1      CALL SHOW_THRUSTERS

 69   1      ACTION_WAIT              /* WAIT FOR  REPRISE */
               DO FOREVER;
 70   2          ACTION = NOT (INPUT(PORT_B));
 71   2          IF ACTION = 2 THEN CALL H_REPRISE;
 73   2          IF ACTION = 4 THEN CALL V_REPRISE;
 75   2         END ACTION_WAIT;

 76   1      END MAIN_DRAGON_MODULE;



MODULE INFORMATION:

     CODE AREA SIZE     = 0181H    385D
     CONSTANT AREA SIZE = 0000H      0D
     VARIABLE AREA SIZE = 0002H      2D
     MAXIMUM STACK SIZE = 000CH     12D
     125 LINES READ
     0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION
```

```
ISIS-II PL/M-86 V2.1 COMPILATION OF MODULE DRAGON_UTILITY
OBJECT MODULE PLACED IN :F1:DRAGUT.OBJ
COMPILER INVOKED BY:  PLM86 :F1:DRAGUT.PLM DEBUG ROM IXREF DATE (02/24/09)


   1          DRAGON_UTILITY: DO;

              /***********************************************************************
                                       OFF-BOARD ABSOLUTE ADDRESSES
              ***********************************************************************/

   2   1      DECLARE (ACTUAL_THRUSTER_ASCII,IDEAL_THRUSTER_ASCII) (24) BYTE AT (0A050H),
                      DUMMY BYTE AT (0A020H);

              /******************** END OFF-BOARD ABSOLUTE ADDRESSES**********************/

   3   1      DECLARE  COUNTER0 LITERALLY '0D0H', COUNTER1 LITERALLY '0D2H',
                      CONTROL LITERALLY '0D6H';     /* SEE PAGE 3-3 IN 86/12 MAN */
   4   1      DECLARE CNTR0MODE LITERALLY '34H',  /* 2 BYTES, MODE 2 */
                      CNTR1MODE LITERALLY '74H';         /* 2 BYTES, MODE 2 */

              /* TO PROGRAM THE 8253 PROGRAMMABLE INTERVAL TIMER OR "PIT" NOTE THAT */
              /* THE INPUT FREQUENCY TO CLK0 IS 1.23 MHZ ==> AN 813 NANOSEC PERIOD  */

   5   1      DECLARE LOW0 LITERALLY '06',     /* COUNTER 0 PERIOD IS 5 MILLESECONDS */
                     HIGH0 LITERALLY '18H';

              /* NOW WE WILL CONNECT OUT0 TO CLK1 BY JUMPER 59-61 INSTEAD OF THE
                DEFAULT CONNECTION, WHICH IS 59-60  */

   6   1      DECLARE LOW1 LITERALLY '5FH',          /* COUNTER 1 PERIOD IS 5 MINUTES */
                     HIGH1 LITERALLY '0EAH';      /* = 5 MIN * 60 (SEC/MIN) / 0.005 SEC -1 IN HEX */


   7   1      DECLARE TIME$LATCH LITERALLY '40H'       /* A COUNTER 1 LATCH  PAGE 3-13 OF 86/12  */
               (LS$TIME$BYTE, MS$TIME$BYTE) BYTE PUBLIC;
   8   1      DECLARE TIME ADDRESS PUBLIC;
   9   1      DECLARE LOW$TIME$BYTE BYTE AT (.TIME), HIGH$TIME$BYTE BYTE AT (.TIME + 1);

  10   1      TIMER$START: PROCEDURE PUBLIC;
  11   2       OUTPUT(CONTROL)=CNTR0MODE;     /* SET COUNTERS 0 & 1 MODES */
  12   2       OUTPUT(CONTROL)=CNTR1MODE;
  13   2       OUTPUT(COUNTER0)=LOW0;       /* INITIALIZE COUNTERS */
  14   2       OUTPUT(COUNTER0)=HIGH0;
  15   2       OUTPUT(COUNTER1)=LOW1;
  16   2       OUTPUT(COUNTER1)=HIGH1;
  17   2       END TIMER$START;

  18   1      CLOCK$READ: PROCEDURE ADDRESS PUBLIC;   /* GETS THE CONTENTS OF COUNTER 1 */
  19   2       OUTPUT(CONTROL)=TIME$LATCH;
  20   2       LOW$TIME$BYTE=INPUT(COUNTER1);
  21   2       HIGH$TIME$BYTE=INPUT(COUNTER1);
  22   2       RETURN TIME;
  23   2       END CLOCK$READ;

  24   1      DECLARE (IDEAL_S$TCNT,TARGET_S$TCNT) INTEGER EXTERNAL;
```

```
                  /* N B  REFERENCES ARE TO ISCB 86/12 9800645A */

 25   1      DECLARE PPI_CONTROL LITERALLY '0CEH', BUS_FREE LITERALLY '0BH';

 26   1      DECLARE SENARIO BYTE AT (6000H);

 27   1      TARGET_LOC  PROCEDURE(ID) INTEGER PUBLIC;
 28   2        DECLARE ID BYTE;   /* ID IS IDENTIFICATION OF H.V. OR X TARG */
 29   2        DECLARE POSITION INTEGER;

 30   2      IF SENARIO = 'A' THEN
 31   2        SCENE_A: DO;
 32   3          IF ID = 1 THEN
 33   3          TARG_H: DO;              /* REPEAT FOR V AND X OF TARGET */
 34   4          END TARG_H;
 35   3          RETURN POSITION;
 36   3        END SCENE_A;

 37   2      IF SENARIO = 'B' THEN       /* ETC, ETC */
 38   2        SCENE_B  DO;
 39   3        END SCENE_B;

 40   2      END TARGET_LOC;

 41   1      INIT_STEPPER  PROCEDURE PUBLIC;
 42   2        IF SENARIO =  A  THEN
 43   2        INIT_A  DO;
 44   3        END INIT_A;

 45   2        IF SENARIO = 'B  THEN    /* ETC, ETC */
 46   2        INIT_B  DO;
 47   3        END INIT_B;

 48   2      END INIT_STEPPER;

 49   1      HX2AS  PROCEDURE (HEX_ADR,ASCII_ADR) PUBLIC;
 50   2        DECLARE (HEX_ADR,ASCII_ADR) POINTER, HEX BASED HEX_ADR INTEGER,
                      ASCII BASED ASCII_ADR (5) BYTE, (N,M) BYTE, REMAINDER INTEGER;

 51   2      HEX = IABS(HEX);
 52   2      DO N=0 TO 4;
 53   3        M = 4-N;
 54   3        REMAINDER = HEX MOD 10 + 30H;
 55   3        ASCII(M) = LOW(UNSIGN(REMAINDER));
 56   3        HEX = HEX/10;
 57   3        END;
 58   2        N=0;

 59   2      DO WHILE ASCII(N) = 30H AND N<5;  /* REPLACE LEADING ZEROES WITH BLANKS */
 60   3        ASCII(N) = 20H;
 61   3        N = N+1;
 62   3        END;
 63   2      END HX2AS;

 64   1      SHOW_THRUSTERS  PROCEDURE PUBLIC;
 65   2        CALL HX2AS(@TARGET_S$TCNT,@ACTUAL_THRUSTER_ASCII);
 66   2         ACTUAL_THRUSTER_ASCII(5)  = 'T';
 67   2         ACTUAL_THRUSTER_ASCII(6)  = 'H';
 68   2         ACTUAL_THRUSTER_ASCII(7)  = 'R';
 69   2         ACTUAL_THRUSTER_ASCII(8)  = 'U';
 70   2         ACTUAL_THRUSTER_ASCII(9)  = 'S';
 71   2         ACTUAL_THRUSTER_ASCII(10) = 'T';
 72   2         ACTUAL_THRUSTER_ASCII(11) = 'E';
 73   2         ACTUAL_THRUSTER_ASCII(12) = 'R';
 74   2         ACTUAL_THRUSTER_ASCII(13) = 'S';
 75   2         ACTUAL_THRUSTER_ASCII(14) = ' ';
 76   2         ACTUAL_THRUSTER_ASCII(15) = 'U';
 77   2         ACTUAL_THRUSTER_ASCII(16) = 'S';
 78   2         ACTUAL_THRUSTER_ASCII(17) = 'E';
 79   2         ACTUAL_THRUSTER_ASCII(18) = 'D';

 80   2       CALL HX2AS(@IDEAL_S$TCNT,@IDEAL_THRUSTER_ASCII);
 81   2         IDEAL_THRUSTER_ASCII(5)  = 'T';
 82   2         IDEAL_THRUSTER_ASCII(6)  = 'H';
 83   2         IDEAL_THRUSTER_ASCII(7)  = 'R';
 84   2         IDEAL_THRUSTER_ASCII(8)  = 'U';
 85   2         IDEAL_THRUSTER_ASCII(9)  = 'S';
 86   2         IDEAL_THRUSTER_ASCII(10) = 'T';
 87   2         IDEAL_THRUSTER_ASCII(11) = 'E';
 88   2         IDEAL_THRUSTER_ASCII(12) = 'R';
 89   2         IDEAL_THRUSTER_ASCII(13) = 'S';
 90   2         IDEAL_THRUSTER_ASCII(14) = ' ';
 91   2         IDEAL_THRUSTER_ASCII(15) = 'I';
 92   2         IDEAL_THRUSTER_ASCII(16) = 'D';
 93   2         IDEAL_THRUSTER_ASCII(17) = 'E';
 94   2         IDEAL_THRUSTER_ASCII(18) = 'A';
 95   2         IDEAL_THRUSTER_ASCII(19) = 'L';
 96   2         IDEAL_THRUSTER_ASCII(20) = 'L';
 97   2         IDEAL_THRUSTER_ASCII(21) = 'Y';

 98   2       END SHOW_THRUSTERS;

 99   1       MISS_COMMENT: PROCEDURE(HEX$ADR, DEC$ADR, DIRECTION) PUBLIC;
100   2       DECLARE(HEX$ADR, DEC$ADR) POINTER,
                     PHRASE BASED DEC$ADR (16) BYTE, (N,DIRECTION) BYTE;

101   2       CALL HX2AS(HEX$ADR,DEC$ADR);

102   2       PHRASE(5) = ' ';
103   2       PHRASE(6) = 'F';
104   2       PHRASE(7) = 'F';
105   2       PHRASE(8) = 'E';
106   2       PHRASE(9) = 'T';
107   2       PHRASE(10) = ' ';
108   2       DO CASE DIRECTION;
109   3         DO;
110   4       PHRASE(11) = 'F';
111   4       PHRASE(12) = 'I';
112   4       PHRASE(13) = 'G';
113   4       PHRASE(14) = 'H';
114   4       PHRASE(15) = 'T';
115   4         END;
116   3         DO;
```

```
117   4        PHRASE(11) = '[illegible]';
118   4        PHRASE(12) = 'E';
119   4        PHRASE(13) = 'F';
120   4        PHRASE(14) = 'T';
121   4          END;
122              DO;
123   4        PHRASE(11) = '[illegible]';
124   4        PHRASE(1[illegible]) [illegible]
125   4        PHRASE(13) = 'G';
126   4        PHRASE(14) = 'H';
127   4          END;
128              DO;
129   4        PHRASE(11) = 'L';
130   4        PHRASE(12) = 'O';
131   4        PHRASE(13) = 'W';
132   4          END;
133              DO;
134   4        PHRASE(11) = 'S';
135   4        PHRASE(12) = 'H';
136   4        PHRASE(13) = 'O';
137   4        PHRASE(14) = 'R';
138   4        PHRASE(15) = 'T';
139   4          END;
140   3        END;
141   2        END MISS_COMMENT;


142   1        SOUND: PROCEDURE (WHAT_KIND) PUBLIC;
143   2          DECLARE WHAT_KIND BYTE, PORT_A LITERALLY '0C8H';
144   2          OUTPUT (PORT_A) = WHAT_KIND;        /* SOUND COMMAND OUTPUT THRU PORT_A */
145   2          OUTPUT (PPI_CONTROL) = 1;      /* SETS PPI PORT_C BIT 0 FOR SOUND CONTROL */
146   2          OUTPUT (PPI_CONTROL) = 1       /* EXTRA ONES FOR 6 MICRO-SEC TIMING */
147   2          OUTPUT (PPI_CONTROL) = 1;
148   2          OUTPUT (PPI_CONTROL) = 0; /* RESETS PPI PORT_C BIT 0 */
149   2          END SOUND;


150   1        ON_BRD_SET: PROCEDURE (SEMA4_PTR, BIT) PUBLIC;
151   2          DECLARE SEMA4_PTR POINTER, BIT BYTE,  BUS_LOCK LITERALLY '0AH';
                 /* N.B. 8255 C-5, WHEN RESET (=0), ASSERTS THE BUS "OVERRIDE." */
                 /* REFER TO PAGE 3-16,-17 FOR PPI PORT_C BIT SET/RESET  */
152   2          DECLARE TEMP BYTE, SEMA4 BASED SEMA4_PTR BYTE;
153   2         TEMP = 1;

154   2         IF BIT = 1 THEN SET_SEMA4

                DO WHILE TEMP = 1;
156   3          FIRST_CHECK: DO WHILE SEMA4 = 1;  /* WAIT HERE TO AVOID REPEATEDLY LOCKING/UNLOCKING BUS */
157   4            END FIRST_CHECK;

158   3         OUTPUT (PPI_CONTROL) = BUS_LOCK;   /* THIS REQUESTS A BUS-LOCK WHEN WE GET CONTROL */
159   3         TEMP = DUMMY;  /* WHEN THIS EXECUTES, WE HAVE CONTROL OF A LOCKED BUS */
160   3         TEMP = SEMA4;   /* SAVE CURRENT BIT OF THE SEMA4 */
161   3         SEMA4 = 1;       /* SEMA4 IS NOW SET. WILL DETERMINE LATER IF WE DID IT */
162   3         OUTPUT(PPI_CONTROL) = BUS_FREE  /* IF SEMA4 WERE OFF-BOARD, COULD REPLACE THESE */
                                               /* LAST FIVE STEPS WITH "LOCKSET(SEMA4_PTR,1)" */
                 /* IF SEMA4 WAS SET ANOTHER MASTER DID IT  SO MUST WAIT TILL USER RESETS IT  */

 163   3          END SET_SEMA4;

 164   2         IF BIT = 0 THEN CLEAR_SEMA4: DO;
 166   3           OUTPUT(PPI_CONTROL) = BUS_LOCK;
 167   3           TEMP = DUMMY;
 168   3           SEMA4 = 0; /* ASSUMES PROGRAM NEVER CLEARS SEMA4 UNLESS SAME PROG SET IT */
 169   3           OUTPUT(PPI_CONTROL) = BUS_FREE;
 170   3           END CLEAR_SEMA4;

 171   2         RETURN;

 172   2          END ON_BRD_SET;


 173   1        PPI_SET  PROCEDURE PUBLIC;
 174   2         DO;
 175   3          DECLARE PPI_MODE LITERALLY '92H';      /* PORTS A & C OUTPUT  B INPUT  */
                                  /* REF PAGES 2-10 & 3-15 */
 176   3          DECLARE SFLAG LITERALLY '0';          /* RESETS PORT "C" BIT 0  */
                 /* WHICH OUTPUTS A "1" FROM 7400; SETTING THE "SOUND-FLAG" TO 8748 */

 177   3          OUTPUT(PPI_CONTROL) = PPI_MODE;   /* ALL PPI OUTPUTS GO LOW INCLUDING
                     PORT-C; BIT-5; WHICH ASSERTS THE MULTI-BUS "OVERRIDE " */
 178   3          OUTPUT(PPI_CONTROL) = BUS_FREE;   /* NEGATES THE MULTI-BUS OVERRIDE  */
 179   3          OUTPUT(PPI_CONTROL) = SFLAG;
 180   3           END;
 181   2         END PPI_SET;

 182   1        TIME_DELAY  PROCEDURE (HOW_LONG) PUBLIC;
 183   2          DECLARE (HOW_LONG, TEST_WORD) WORD;
 184   2         LOOPA  DO WHILE HOW_LONG <> 0;
 185   3            TEST_WORD = 39H;
 186   3            LOOPB  DO WHILE TEST_WORD <> 0;
 187   4               TEST_WORD = TEST_WORD -1;
 188   4              END LOOPB;
 189   3           HOW_LONG = HOW_LONG - 1;
 190   3          END LOOPA;
 191   2         END TIME_DELAY;

 192   1        END DRAGON_UTILITY;



MODULE INFORMATION

     CODE AREA SIZE     = 0475H   1141D
     CONSTANT AREA SIZE = 0000H      0D
     VARIABLE AREA SIZE = 000EH     14D
     MAXIMUM STACK SIZE = 001AH     26D
     264 LINES READ
     0 PROGRAM ERROR(S)

END OF PL/M-86 COMPILATION
```

```
ISIS-II MCS-86 MACRO ASSEMBLER V2 1 ASSEMBLY OF MODULE RDRSB
OBJECT MODULE PLACED IN  F1 DRAGXF OBJ
ASSEMBLER INVOKED BY   ASM86  F1 DRAGXF SRC DEBUG DATE (02/24/09)


LOC  OBJ                  LINE     SOURCE

                             1     ;THIS PROGRAM, STARTED SEPT 5, 1979, READS DATA FROM THE RETICON RSB6020
                             2     ; INTERFACE BOARD INTO THE 86/12 MEMORY  REFERENCES ARE:
                             3     ;        1  RSB6020 OPERATING INSTRUCTIONS, MARCH 9, 1979, EG&G RETICON
                             4     ;           SUNNYVALE, CALIFORNIA
                             5     ;        2  MCS-86 ASSEMBLE LANGUAGE REFERENCE MANUAL, #9800640A,
                             6     ;           INTEL CORP, SANTA CLARA, CALIFORNIA
                             7     ;
                             8     ;IT IS BEING CLEANED UP A BIT FEB 20, 1981
                             9
                            10     ;EQUATES AT TOP OF PROGRAM PER P  8-1, REF 2
                            11
                            12            NAME    RDRSB
                            13     DGROUP GROUP   DATA, STACK, SBC_REGS, RSB_REGS
                            14     CGROUP GROUP   CODE
                            15
  0001                      16     INMSK   EQU    01        ;SET UP FOR CAMERA 1 ONLY  SEE P  19, REF 1
  0064                      17     LINES   EQU    100
  0010                      18     ENDFR   EQU    10H       ;MASK FOR THE 2-TO-THE-4TH BIT, P  45, REF 1
                            19
                            20     ASSUME SS DGROUP, CS CGROUP, DS DGROUP, ES DGROUP
                            21
----                        22     SBC_REGS SEGMENT COMMON           ;NOTE THAT "COMMON" FILES MUST BE
                            23                                       ;COMMON IN ALL MODULES, I E  CAN'T BE
0000 (800                   24     SBCREG  DB      800 DUP (?)       ;"AT" IN ONE AND "COMMON" IN ANOTHER
     ??
     )

                            25                                       ;THEY DO NOT, HOWEVER, HAVE TO BE POINTED
                            26                                       ;TO BY THE SAME SEGMENT REGISTER IN BOTH
                            27                                       ;MODULES, NOR DO THEY HAVE TO
                            28                                       ;OF THE SAME LENGTH
0320 (6                     29     PARTLY_OFF      DB      6 DUP (?)       ; INITIALEZE = 0
     ??
     )

0326 (6                     30     LOCATIONS       DB      6 DUP (?)       ; INITIALIZE = 1
     ??
     )

----                        31     SBC_REGS ENDS
                            32
----                        33     RSB_REGS SEGMENT AT 0EA00H        ;BASE ADDRESS OF RETICON BOARD IS AEA00H
0000 (512                   34     RSBDTA  DB      200H DUP (?)
     ??
     )

0200 (1                     35     STAT1   DB      1 DUP (?)         ;THIS FORM IS NECESSARY TO AVOID LOADING ERRORS
     ??
     )

0201 (11                    36     STAT2   DB      0BH DUP (?)
     ??
     )

020C (2                     37     RESET   DB      2 DUP(?)
     ??
```

```
LOC  OBJ                  LINE     SOURCE

020E (1                     38     CNFG15  DB      1 DUP (?)
     ??
     )
020F (1                     39     PROCOM  DB      1 DUP (?)
     ??
     )
----                        40     RSB_REGS ENDS
                            41
----                        42     STACK   SEGMENT STACK   'STACK'
0000 (10                    43             DW      10 DUP (?)
     ????
     )
0014                        44     STKTOP  LABEL   WORD
----                        45     STACK   ENDS
                            46
                            47
----                        48     CODE    SEGMENT PUBLIC  'CODE'
                            49
                            50             PUBLIC  INIT1
0000                        51     INIT1   PROC    NEAR            ;INITIALIZATION OF RSB 6020 INTERFACE BOARD
0000 A20C02          R      52     RTINIT  MOV     RESET,AL        ;RESET IS A "DUMMY" REGISTER, ALL IT NEEDS
0003 C6060E0201      R      53             MOV     CNFG15,INMSK    ;IS THE "MWTC/" PULSE FROM P1 #20
0008 A20F02          R      54             MOV     PROCOM,AL       ;PROCOM IS ALSO A "DUMMY" REGISTER
000B C3                     55             RET
                            56     INIT1   ENDP
                            57
                            58             PUBLIC  RD_RAST
000C                        59     RD_RAST PROC    NEAR
000C A20F02          R      60     LAST    MOV     PROCOM,AL       ;WILL WAIT FOR LAST RASTER LINE
000F A00002          R      61     WTLP0   MOV     AL,STAT1        ;FROM HERE TO CHECK IS ST'D NUCMCY
0012 D0E0                   62             SHL     AL,1
0014 7303                   63             JNB     WTLP3
0016 A20F02          R      64     PSPR0   MOV     PROCOM,AL
0019 A00002          R      65     WTLP3   MOV     AL,STAT1
001C D0E0                   66             SHL     AL,1
001E 7206                   67             JB      CHECK
0020 D0E0                   68             SHL     AL,1
0022 72F5                   69             JB      WTLP3
0024 EBF0                   70             JMP     PSPR0
0026 A00300          R      71     CHECK   MOV     AL,RSBDTA[3]
0029 2410                   72             AND     AL,ENDFR
002B 74DF                   73             JZ      LAST
                            74
                            75     ;HAVING FOUND THE LAST LINE OF THE FRAME, WE WILL TRANSFER THE RETICON
                            76     ;DATA, LINE-BY-LINE THE THE INTEL 86/12 BOARD. EACH LINE TRANSFERRED STARTS
                            77     ;WITH A "NEW COMMAND CYCLE" AS PER PAGE 40 OF REFERENCE 1 ^
                            78
002D B364                   79             MOV     BL,LINES                ;WILL DECREMENT FROM 100 TO ZERO
002F BF0000          R      80             MOV     DI,OFFSET DGROUP:SBCREG
                            81
0032 A20F02          R      82     NUCMCY  MOV     PROCOM,AL       ;AGAIN, PROCOM IS A DUMMY
0035 A00002          P      83     WTLP1   MOV     AL,STAT1
0038 D0E0                   84             SHL     AL,1
003A 7303                   85             JNB     WTLP
```

```
LOC  OBJ              LINE    SOURCE

003C A20F02      R       86   PSPR:   MOV     PROCOM,AL
003F A00002      R       87   WTLP:   MOV     AL,STAT1
0042 D0E0                88           SHL     AL,1
0044 7206                89           JB      OTLP
0046 D0E0                90           SHL     AL,1
0048 72F5                91           JB      WTLP
004A EBF0                92           JMP     PSPR
                         93
                         94   ;NOW TRANSFER A SINGLE LINE OF RETICON RSB 6020 DATA TO THE INTEL 86 BOARD.
                         95
004C A00000      R       96   OTLP:   MOV     AL,RSBDTA                ;TRANSITIONS IN THE LINE
004F 32E4                97           XOR     AH,AH
0051 40                  98           INC     AX                       ;NUMBER OF WORDS TO XFER
0052 D1E0                99           SHL     AX,1                     ;X2 NUMBER OF BYTES TO XFER
0054 8BC8               100           MOV     CX,AX
0056 BE0000      R      101           MOV     SI,OFFSET DGROUP:RSBDTA
0059 F3                 102   REP     MOVS    BYTE PTR SBCREG [DI], BYTE PTR RSBDTA [SI]
005A A4
005B FECB               103           DEC     BL
005D 75D3               104           JNZ     NUCMCY
                        105
005F C3                 106           RET
                        107
                        108   RD_RAST ENDP
                        109
----                    110   CODE    ENDS
                        111
                        112           END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

APPENDIX C

COMPUTER GRAPHICS AND VIDEO SUBSYSTEM PROGRAMS

```
SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.0 ASSEMBLY OF MODULE MATROX
OBJECT MODULE PLACED IN :F2:MATROX.OBJ
INVOCATION LINE CONTROLS:  DEBUG


LOC  OBJ                  LINE     SOURCE

                            1      NAME MATROX
                            2
                            3      CGROUP  GROUP   CODE
                            4
                            5      DATA_GROUP      GROUP   DGROUP,DATA_SEG,GAE_SEG
                            6      ASSUME  CS:CGROUP, DS:DATA_GROUP
                            7      PUBLIC  BACKGROUND, MATROX_START_UP,START_BIT,X_Y,DATA_READY,ERROR
                            8      PUBLIC  XCNT,YCNT,BAD_MISS,FIRE_BIRD,THRUSTER_FIRE
                            9      EXTRN   SMOKE_SET: NEAR,GAE_GT_FOV:NEAR,GRAPH_VREP:NEAR
                           10      EXTRN   GRAPH_GAE_POINT: NEAR,GRAPH_HREP:NEAR
                           11      EXTRN   SMOKE_CHECK  NEAR
                           12      EXTRN   GRAPH1  NEAR
                           13      EXTRN   SMOKE_START_UP:NEAR
                           14      EXTRN   GAE_START_UP:NEAR
                           15      EXTRN   COUT:NEAR
                           16      EXTRN   CIN:NEAR
                           17      EXTRN   HIT_EXPLOSION:NEAR,GROUND_EXPLOSION:NEAR
                           18      EXTRN   TANK_INIT:NEAR,TANK_KILLED:NEAR
                           19      EXTRN   USART_SET_UP_FOR_ADM:NEAR
                           20      EXTRN   DELAY_3:NEAR  ;160MS DELAY FOR CLEARING RETRO_GRAPHICS SCREEN
                           21
----                       22      DGROUP  SEGMENT PUBLIC 'DATA'
----                       23      DGROUP  ENDS
                           24
----                       25      DATA_SEG        SEGMENT PUBLIC
                           26      EXTRN   H_REP_FLAG:BYTE,V_REP_FLAG:BYTE
----                       27      DATA_SEG        ENDS
                           28
----                       29      GAE_SEG         SEGMENT  PUBLIC
                           30      EXTRN   ELAPSED_TIME:WORD
----                       31      GAE_SEG         ENDS
                           32
                           33
  0014                     34      XREG    EQU     14H
  0016                     35      YREG    EQU     16H
  0010                     36      GSCALE  EQU     10H
  0012                     37      SCROLL  EQU     12H
  0012                     38      FLAGS   EQU     12H
  0014                     39      ERASE   EQU     14H
                           40
----                       41      DATA_SEG        SEGMENT PUBLIC
                           42
0000 (1                    43      XCNT            DB      1 DUP(?)
      ??
      )
0001 (1                    44      YCNT            DB      1 DUP(?)
      ??
      )
0002 (1                    45      XCNT_OLD        DB      1 DUP(?)
      ??
LOC  OBJ                  LINE    SOURCE

       )
0003  (1                   46     YCNT_OLD        DB      1 DUP(?)
       ??
       )
0004  (1                   47     XCNT_SAVE       DB      1 DUP(?)
       ??
       )
0005  (1                   48     YCNT_SAVE       DB      1 DUP(?)
       ??
       )
0006  (1                   49     GSCALE_VAL      DB      1 DUP(?)
       ??
       )
0007  (1                   50     GSCALE_NUM      DB      1 DUP(?)
       ??
       )
0008  (1                   51     THRUSTER_FIRE   DB      1 DUP(?)
       ??
       )
0009  (1                   52     SIZ             DW      1 DUP(?)
       ????
       )
000B  (1                   53     XMIN            DW      1 DUP(?)
       ????
       )
000D  (1                   54     YMIN            DW      1 DUP(?)
       ????
       )
000F  (1                   55     XMAX            DW      1 DUP(?)
       ????
       )
0011  (1                   56     YMAX            DW      1 DUP(?)
       ????
       )
0013  (1                   57     XMAX_TMP        DW      1 DUP(?)
       ????
       )
0015  (1                   58     XMIN_TMP        DW      1 DUP(?)
       ????
       )
0017  (1                   59     TMP_WRD         DW      1 DUP(?)
       ????
       )
0019  (1                   60     ONE_THIRD       DW      1 DUP(?)
       ????
       )
001B  (1                   61     TWO_THIRDS      DW      1 DUP(?)
       ????
       )
001D  (1                   62     THREE           DB      1 DUP(?)
       ??
       )
001E  (1                   63     COUNT_EM        DB      1 DUP(?)
       ??
       )
```

```
LOC  OBJ                  LINE    SOURCE

001F (1                     64    BACKGROUND      DB      1 DUP(?)
      ??
      )
0020 (1                     65    GSCALE_SAVE     DB      1 DUP(?)
      ??
      )
0021 (1                     66    COUNT           DB      1 DUP(?)
      ??
      )
0022 (1                     67    BLINK_COUNT     DB      1 DUP(?)
      ??
      )
0023 (1                     68    REPLAY          DB      1 DUP(?)
      ??
      )
0024 (1                     69    ERROR           DB      1 DUP(?)
      ??
      )
0025 (1                     70    ERROR_MESG_FLAG DB      1 DUP(?)
      ??
      )
0026 (1                     71    GRND_BIRD_FLAG  DB      1 DUP(?)
      ??
      )
0027 (1                     72    HIT_FLAG        DB      1 DUP(?)
      ??
      )
0028 (1                     73    DIST_FRO_TGT_FLAG DB    1 DUP(?)
      ??
      )
0029 (1                     74    RESULTS_FLAG    DB      1 DUP(?)
      ??
      )
002A (1                     75    THRUSTERS_FLAG  DB      1 DUP(?)
      ??
      )
002B (1                     76    TWO             DW      1 DUP(?)
      ????
      )
002D (1                     77    YANG_SCALED     DW      1 DUP(?)
      ????
      )
002F (1                     78    ZANG_SCALED     DW      1 DUP(?)
      ????
      )
0031 (1                     79    YANG2_OLD       DW      1 DUP(?)
      ????
      )
0033 (1                     80    ZANG2_OLD       DW      1 DUP(?)
      ????
      )
0035 (1                     81    YANG2_SAVE      DW      1 DUP(?)
      ????
      )
0037 (1                     82    ZANG2_SAVE      DW      1 DUP(?)
LOC  OBJ                  LINE     SOURCE

     ????
     )
0039 (1                     83     YANG2          DW       1 DUP(?)
     ????
     )
003B (1                     84     ZANG2          DW       1 DUP(?)
     ????
     )
003D (1                     85     FLAG           DB       1 DUP(?)
     ??
     )
003E (1                     86     POP_LEVEL      DB       1 DUP(?)
     ??
     )
003F (1                     87     BORE_SIGHT     DB       1 DUP(?)
     ??
     )
0040 (1                     88     START_UP_BYTE  DB       1 DUP(?)
     ??
     )
                            89
----                        90     DATA_SEG       ENDS
                            91
                            92
----                        93     XFER_SEG         SEGMENT        AT 0600H
0000 (1                     94     START_BIT      DB       1 DUP(?)
     ??
     )
0001 (1                     95     X_Y            DW       1 DUP(?)
     ????
     )
0003 (1                     96     DATA_READY     DB       1 DUP(?)
     ??
     )
0004 (1                     97     BAD_MISS       DB       1 DUP(?)
     ??
     )
0005 (1                     98     OFFSET_X       DB       1 DUP(?)
     ??
     )
0006 (1                     99     OFFSET_Y       DB       1 DUP(?)
     ??
     )
0007 (9                    100     DUMMYS         DB       9 DUP(?)
     ??
     )
0010 (1                    101     YANG           DW       1 DUP(?)
     ????
     )
0012 (1                    102     ZANG           DW       1 DUP(?)
     ????
     )
0014 (1                    103     BIRD_DTA_RDY   DB       1 DUP(?)
     ??
     )
```

```
LOC  OBJ                 LINE     SOURCE

0015 (1                   104     FIRE_BIRD       DB       1 DUP(?)
      ??
      )
0016 (1                   105     BIRD_HIT        DB       1 DUP(?)
      ??
      )
0017 (1                   106     BIRD_MISSES     DB       1 DUP(?)
      ??
      )
0018 (1                   107     H_REP_REQ       DB       1 DUP(?)
      ??
      )
0019 (1                   108     H_REP_GO        DB       1 DUP(?)
      ??
      )
001A (1                   109     V_REP_REQ       DB       1 DUP(?)
      ??
      )
001B (1                   110     V_REP_GO        DB       1 DUP(?)
      ??
      )
001C (1                   111     GRND_BIRD       DB       1 DUP(?)
      ??
      )
001D (1                   112     END_OF_RPV      DB       1 DUP(?)
      ??
      )
001E (1                   113     HIT_SHORT       DB       1 DUP(?)
      ??
      )
001F (1                   114     DUMMY_ALSO      DB       1 DUP(?)
      ??
      )
0020 (16                  115     H_MIS_ASCII     DB       16 DUP(?)
      ??
      )
0030 (16                  116     V_MIS_ASCII     DB       16 DUP(?)
      ??
      )
0040 (16                  117     DIS_FRO_TGT     DB       16 DUP(?)
      ??
      )
0050 (24                  118     ACTUAL_THRUST   DB       24 DUP(?)
      ??
      )
0068 (24                  119     IDEAL_THRUST    DB       24 DUP(?)
      ??
      )
----                      120     XFER_SEG        ENDS
                          121
----                      122     STACK_SEG       SEGMENT STACK  'STACK'
0000 (64                  123                     DW      64D DUP(?)
      ????
      )
0080                      124     STKTOP  LABEL   WORD
```

```
LOC  OBJ                LINE     SOURCE

----                     125     STACK_SEG       ENDS
                         126
----                     127     CODE    SEGMENT PUBLIC 'CODE'
                         128
0000 47524F554E4420      129     MESSAGE         DB      'GROUND IMPACT',350
     494D50414354
000D 1D
000E                     130     FIN_OF_MESSAGE  LABEL   WORD
                         131
000E 4D495353494C45      132     MESSAGE2        DB      'MISSILE POSITION EXCEEDED GRAPH BOUNDS',350
     20504F53495449
     4F4E2045584345
     45444544204752
     41504820424F55
     4E4453
0034 1D
0035                     133     FIN_OF_MESSAGE2 LABEL   WORD
                         134
0035 54485255535445      135     MESSAGE3        DB      'THRUSTERS: '
     52533A20
0040 02                  136     END_OF_MESSAGE3 DB      WORD
                         137
0041 B80006              138     DRIVER:         MOV     AX,XFER_SEG
0044 8EC0                139                     MOV     ES,AX
0046 B8----         R    140                     MOV     AX,STACK_SEG
0049 8ED0                141                     MOV     SS,AX
004B B80000         R    142                     MOV     AX,OFFSET STKTOP
004E 8BE0                143                     MOV     SP,AX
0050 B8----         R    144                     MOV     AX,DATA_GROUP
0053 8ED8                145                     MOV     DS,AX
0055 26C606000000        146                     MOV     ES:START_BIT,0
005B 803E400001     R    147                     CMP     START_UP_BYTE,1
0060 740C                148                     JE      NOT_START_UP
0062 26C606050000        149                     MOV     ES:OFFSET_X,0
0068 26C606060000        150                     MOV     ES:OFFSET_Y,0
006E C606400001     R    151     NOT_START_UP:   MOV     START_UP_BYTE,1
0073 C606230000     R    152                     MOV     REPLAY,0
0078 C7062B000200   R    153                     MOV     TWO,2
007E C606280000     R    154                     MOV     DIST_FRO_TGT_FLAG,0
0083 C606260000     R    155                     MOV     GRND_BIRD_FLAG,0
0088 C606270000     R    156                     MOV     HIT_FLAG,0
008D C606290000     R    157                     MOV     RESULTS_FLAG,0
0092 C6063F0000     R    158                     MOV     BORE_SIGHT,0
0097 C606240000     R    159                     MOV     ERROR,0
009C C606250000     R    160                     MOV     ERROR_MESG_FLAG,0
00A1 C6062E0000     R    161                     MOV     GRND_BIRD_FLAG,0
00A6 C606270000     R    162                     MOV     HIT_FLAG,0
00AB C6062A0000     R    163                     MOV     DIST_FRO_TGT_FLAG,0
00B0 E80000         E    164                     CALL    USART_SET_UP_FOR_ADM
00B3 E80000         E    165                     CALL    USART_SET_UP_FOR_ADM
00B6 E80000         E    166                     CALL    SMOKE_START_UP
00B9 E80000         E    167                     CALL    GAE_START_UP ;INITIALIZES THE GAE PROGRAM
00BC E8C801              168                     CALL    MATROX_START_UP  ;CLEARS MATROX SCREEN
00BF BAD800              169                     MOV     DX,0D8H
00C2 B01D                170                     MOV     AL,350
```

```
LOC  OBJ                 LINE     SOURCE

00C4 E80000         E     171                      CALL    COUT
00C7 B013                 172                      MOV     AL,310
00C9 E80000         E     173                      CALL    COUT
00CC E80000         E     174                      CALL    DELAY_3   ;DELAY 160MS
00CF B00C                 175                      MOV     AL,150
00D1 E80000         E     176                      CALL    COUT
00D4 B018                 177                      MOV     AL,0300
00D6 E80000         E     178                      CALL    COUT
00D9 B01A                 179                      MOV     AL,320  ;CLEAR ADM-3 SCREEN
00DB E80000         E     180                      CALL    COUT
00DE E80000         E     181                      CALL    TANK_INIT  ; ALLOWS INST. TO SELECT FROM MENU
00E1 B80006               182                      MOV     AX,XFER_SEG   ;THIS & THE NEXT INST. ARE A CLUGE
                          183                                            ;TO GUARD THE ES IF NECESSARY???
00E4 8EC0                 184                      MOV     ES,AX
00E6 B01A                 185                      MOV     AL,320
00E8 BAD800               186                      MOV     DX,0D8H
00EB E80000         E     187                      CALL    COUT     ;CLEAR ADM-3 SCREEN
00EE E80000         E     188                      CALL    GRAPH1 ;DRAWS AND CLEARS ENVELOPES ON ADM-3 SCREEN
00F1 26C606000001         189                      MOV     ES:START_BIT,1
00F7 E86007               190     WAY_UP_HERE:     CALL    CHECK_FOR_B     ;BORESIGHT CHECK
00FA 80FC01               191                      CMP     AH,1
00FD 7511                 192                      JNE     DO_IT_OVER
00FF 3C02                 193                      CMP     AL,0002H         ;CTRL-B ??
0101 750D                 194                      JNE     DO_IT_OVER
0103 BAD800               195                      MOV     DX,0D8H
0106 B042                 196                      MOV     AL,1020       ;OUTPUT A 'B' TO ADM-3
0108 E80000         E     197                      CALL    COUT
010B C6063F0001     R     198                      MOV     BORE_SIGHT,1
0110 26803E030001         199     DO_IT_OVER:      CMP     ES:DATA_READY,1  ;DATA_READY???
0116 7520                 200                      JNE     CHK_FOR_RPV
0118 803E3F0001     R     201                      CMP     BORE_SIGHT,1
011D 7508                 202                      JNE     BEYOND
011F E84507               203                      CALL    CALIBRATE        ;CALIBRATE FROM FIRST DATA POINTS
0122 C6063F0000     R     204                      MOV     BORE_SIGHT,0
0127 26803E040001         205     BEYOND           CMP     ES:BAD_MISS,1    ;BAD_MISS???
012D 7506                 206                      JNE     DO_IT
012F E80000         E     207                      CALL    GAE_GT_FOV
0132 E94601               208                      JMP     OVER2    ;DO NOT GET NEW DATA IF BAD_MISS
0135 E9FF00               209     DO_IT:           JMP     DO_IT_LONG
                          210
0138 26803E1A0001         211     CHK_FOR_RPV      CMP     ES:V_REP_REQ,1
013E 7423                 212                      JE      V_REP_SHORT
0140 26803E180001         213                      CMP     ES:H_REP_REQ,1
0146 741E                 214                      JE      H_REP_SHORT
0148 26803E1E0001         215                      CMP     ES:HIT_SHORT,1
014E 7436                 216                      JE      DIST_FROM_TGT
0150 26803E1D0001         217                      CMP     ES:END_OF_RPV,1
0156 744C                 218                      JE      ERROR_MSG_CHK
0158 26803E170001         219                      CMP     ES:BIRD_MISSES,1
015E 741C                 220                      JE      TEST1
0160 EB0790               221                      JMP     BACK
0163 E9AF00               222     V_REP_SHORT:     JMP     V_REP
0166 E98A00               223     H_REP_SHORT:     JMP     H_REP
                          224
0169 26803E160001         225     BACK:            CMP     ES BIRD_HIT,1
```

```
LOC  OBJ                 LINE     SOURCE

016F 7427                 226                     JE      PRNT_HIT
0171 26803E1C0001         227                     CMP     ES:GRND_BIRD,1
0177 747?                 228                     JE      PRNT_GRND_BRD
0179 E976FF               229                     JMP     WAY_UP_HERE
                          230
017D 26803E1C0001         231     TEST1:          CMP     ES:GRND_BIRD,1 ;WAIT FOR END OF FLIGHT BEFORE
0182 7406                 232                     JE      PRINT_RESULT    ;PRINTING MESSAGES
0184 EBE3                 233                     JMP     BACK
                          234
0186 E8FE00               235     DIST_FROM_TGT:  CALL    MATROX_START_UP
0189 E80000         E     236                     CALL    GROUND_EXPLOSION
018C E9B704               237                     JMP     DIST_FRO_TGT
018F E8F500               238     PRINT_RESULT:   CALL    MATROX_START_UP
0192 E80000         E     239                     CALL    GROUND_EXPLOSION
0195 E9B804               240                     JMP     PRINT_RESULTS
0198 E8EC00               241     PRNT_HIT        CALL    MATROX_START_UP
019B E80000         E     242                     CALL    HIT_EXPLOSION
019E E80000         E     243                     CALL    TANK_KILLED
01A1 E95405               244                     JMP     PRINT_HIT
                          245
01A4 803E280001     R     246     ERROR_MSG_CHK:  CMP     DIST_FRO_TGT_FLAG,1
01A9 7506                 247                     JNE     SKIP
01AB 26C6061E0001         248                     MOV     ES:HIT_SHORT,1
01B1 803E260001     R     249     SKIP:           CMP     GRND_BIRD_FLAG,1
01B6 7506                 250                     JNE     SKIP2
01B8 26C6061C0001         251                     MOV     ES:GRND_BIRD,1
01BE 803E270001     R     252     SKIP2:          CMP     HIT_FLAG,1
01C3 7506                 253                     JNE     SKIP3
01C5 26C606160001         254                     MOV     ES:BIRD_HIT,1
01CB 803E290001     R     255     SKIP3:          CMP     RESULTS_FLAG,1
01D0 7506                 256                     JNE     SKIP4
01D2 26C606170001         257                     MOV     ES:BIRD_MISSES,1
01D8 803E240001     R     258     SKIP4:          CMP     ERROR,1
01DD 740E                 259                     JE      ERROR_MESG_LONG
01DF 26C6061D0000         260                     MOV     ES:END_OF_RPV,0
01E5 C606000000     E     261                     MOV     H_REP_FLAG,0
01EA E923FF               262                     JMP     DO_IT_OVER
01ED E91A06               263     ERROR_MESG_LONG  JMP    ERROR_MESG
                          264
01F0 E9C905               265     PRNT_GRND_BRD   JMP     PRINT_GRND_BIRD
                          266
01F3 C60623000l     R     267     H_REP           MOV     REPLAY,1
01F8 26C606180000         268                     MOV     ES:H_REP_REQ,0
01FE E80000         E     269                     CALL    GRAPH_HREP
0201 E80000         E     270                     CALL    GAE_START_UP
0204 E89A00               271                     CALL    MATROX_START_UP
0207 C606000001     E     272                     MOV     H_REP_FLAG,1
020C 26C606190001         273                     MOV     ES:H_REP_GO,1
0212 E9FEFE               274                     JMP     DO_IT_OVER
                          275
0215 C60623000l     R     276     V_REP           MOV     REPLAY,1
021A 26C6061A0000         277                     MOV     ES:V_REP_REQ,0
0220 E80000         E     278                     CALL    GRAPH_VREP
0223 E80000         E     279                     CALL    GAE_START_UP
0226 E85E00               280                     CALL    MATROX_START_UP ;THIS MIGHT BE "MATROX_CANCELL"
```

```
LOC  OBJ                 LINE     SOURCE

0229 C606000001     E     281                        MOV      V_REP_FLAG,1
022F 26C6061E0001         282                        MOV      ES:V_REP_GO,1
0234 E9D9FE               283                        JMP      DO_IT_OVER
                          284
0237 A00000         R     285     DO_IT_LONG         MOV      AL,XCNT
023A A20200         R     286                        MOV      XCNT_OLD,AL
023D A00100         R     287                        MOV      AL,YCNT
0240 A20300         R     288                        MOV      YCNT_OLD,AL
0243 A13900         R     289                        MOV      AX,YANG2
0246 A33100         R     290                        MOV      YANG2_OLD,AX
0249 A13B00         R     291                        MOV      AX,ZANG2
024C A33300         R     292                        MOV      ZANG2_OLD,AX
024F 268B160100           293                        MOV      DX,ES:X_Y    ;THIS INST GETS XCNTR,YCNTR IN ONE GULP.
0254 88160000       R     294                        MOV      XCNT,DL
0258 88360100       R     295                        MOV      YCNT,DH
025C 26A11000             296                        MOV      AX,ES:YANG
0260 A33900         R     297                        MOV      YANG2,AX
0263 26A11200             298                        MOV      AX,ES:ZANG
0267 A33B00         R     299                        MOV      ZANG2,AX
026A 26C606070000         300                        MOV      ES:DATA_READY,0
0270 26803E040001         301                        CMP      ES:BAD_MISS,1
0276 7403                 302                        JE       OVER2
0278 E80000         E     303                        CALL     GRAPH_GAE_POINT   ;PLOTS DOTS ON ADM-3.
027B E84D00               304     OVER2              CALL     START    ;PLOTS ROCKET AND SMOKE ON MATROX.
027E 26C606040000         305                        MOV      ES:BAD_MISS,0
0284 E989FE               306                        JMP      CO_IT_OVER
                          307
0287 C606070009     R     308     MATROX_START_UP:   MOV      GSCALE_NUM,09D
028C A00700         R     309                        MOV      AL,GSCALE_NUM
028F A20600         R     310                        MOV      GSCALE_VAL,AL
0292 C70609000900   R     311                        MOV      SIZ,09D
0298 C6063E000A     R     312                        MOV      POP_LEVEL,10D
029D C6061D0003     R     313                        MOV      THREE,3D
02A2 C6063D0000     R     314                        MOV      FLAG,0
02A7 C6061F0000     R     315                        MOV      BACKGROUND,0
02AC C606210000     R     316                        MOV      COUNT,0
02B1 C6061E0000     R     317                        MOV      COUNT_EM,0
02B6 B000                 318                        MOV      AL,0
02B8 E612                 319                        OUT      SCROLL,AL
02BA A01F00         R     320                        MOV      AL,BACKGROUND
02BD E610                 321                        OUT      GSCALE,AL
02BF E414                 322                        IN       AL,ERASE
02C1 E412                 323     WAIT1              IN       AL,FLAGS
02C3 2401                 324                        AND      AL,1
02C5 74FA                 325                        JZ       WAIT1
02C7 C3                   326                        RET
                          327
02C8 E95401               328     STILL_SMOKE_SHORT:          JMP      STILL_SMOKE
02CB 803E230001     R     329     START              CMP      REPLAY,1
02D0 7408                 330                        JE       FIN_OF_MAT
02D2 813E0000F000   E     331                        CMP      ELAPSED_TIME,240D
02D8 7603                 332                        JNA      FIN_OF_MAT
02DA EB0490               333                        JMP      TO
02DD E9C601               334     FIN_OF_MAT         JMP      FIN_OF_MATROX
02E0 A00700         R     335     TO                 MOV      AL,GSCALE_NUM
```

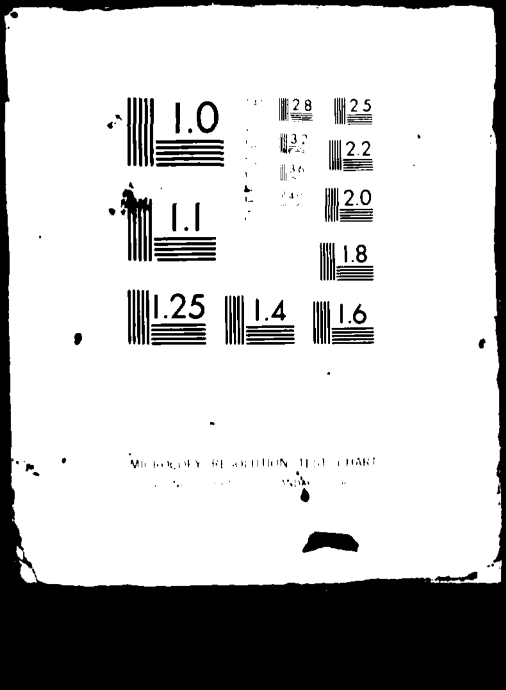
AD-A107 461 ARMY PROJECT MANAGER FOR TRAINING DEVICES ORLANDO FL F/G 19/5
SIMULATED TANK ANTI-ARMOR GUNNERY SYSTEM (STAGS).(U)
NOV 81 A MARSHALL, B SHAW, H TOWLE
UNCLASSIFIED PM TRADE-RE-0010 NL

2 of 2

END
DATE
FILMED
12 81
DTIC

2 OF 2

AD-A107481



```
LOC  OBJ                LINE     SOURCE

02E3 A20600         R    336                    MOV    GSCALE_VAL,AL
02E6 E80000         E    337                    CALL   SMOKE_CHECK
02E9 80FC01              338                    CMP    AH,1
02EC 75DA                339                    JNE    STILL_SMOKE_SHORT
02EE A13900         R    340                    MOV    AX,YANG2
02F1 A33500         R    341                    MOV    YANG2_SAVE,AX
02F4 A13B00         R    342                    MOV    AX,ZANG2
02F7 A33700         R    343                    MOV    ZANG2_SAVE,AX
02FA A00600         R    344                    MOV    AL,GSCALE_VAL
02FD A22000         R    345                    MOV    GSCALE_SAVE,AL
0300 C606060000     R    346                    MOV    GSCALE_VAL,0
                         347
0305 A00000         R    348                    MOV    AL,XCNT
0308 A20400         R    349                    MOV    XCNT_SAVE,AL
030B A00200         R    350                    MOV    AL,XCNT_OLD
030E A20000         R    351                    MOV    XCNT,AL
                         352
0311 A00100         R    353                    MOV    AL,YCNT
0314 A20500         R    354                    MOV    YCNT_SAVE,AL
0317 A00300         R    355                    MOV    AL,YCNT_OLD
031A A20100         R    356                    MOV    YCNT,AL
                         357
031D A13B00         R    358                    MOV    AX,ZANG2
0320 A33700         R    359                    MOV    ZANG2_SAVE,AX
0323 A13300         R    360                    MOV    AX,ZANG2_OLD
0326 A33B00         R    361                    MOV    ZANG2,AX
                         362
0329 A13900         R    363                    MOV    AX,YANG2
032C A33500         R    364                    MOV    YANG2_SAVE,AX
032F A13100         R    365                    MOV    AX,YANG2_OLD
0332 A33900         R    366                    MOV    YANG2,AX
                         367
0335 E86F01              368                    CALL   OCT_DRAW
0338 A02000         R    369                    MOV    AL,GSCALE_SAVE
033B A20600         R    370                    MOV    GSCALE_VAL,AL
                         371
033E A00400         R    372                    MOV    AL,XCNT_SAVE  ;THESE NEXT 4 INST. RESTORE XCNT,YCNT
0341 A20000         R    373                    MOV    XCNT,AL       ;AFTER AN ERASE CYCLE.
0344 A00500         R    374                    MOV    AL,YCNT_SAVE
0347 A20100         R    375                    MOV    YCNT,AL
                         376
034A A13500         R    377                    MOV    AX,YANG2_SAVE ;THESE NEXT 4 INST RESTORE YANG2,ZANG2
034D A33900         R    378                    MOV    YANG2,AX      ;AFTER AN ERASE CYCLE.
0350 A13700         R    379                    MOV    AX,ZANG2_SAVE
0353 A33B00         R    380                    MOV    ZANG2,AX
0356 E80300              381                    CALL   S_P_AND_GSCALE_SET
0359 E9BE00              382                    JMP    SIZ_IS_SET2
                         383      ;*********************************************************************
035C A10000         E    384      S_P_AND_GSCALE_SET:    MOV    AX,ELAPSED_TIME
035F 3DFA01              385                    CMP    ,506D  ; WAS 500
0362 7513                386                    JNE    WAIT0    ;START THE SIZ,POP,AND GSCALE SETTING ROUTINE
0364 C70609000800   R    387                    MOV    SIZ,8D
036A C606070008     R    388                    MOV    GSCALE_NUM,8D
036F C6063E0008     R    389                    MOV    POP_LEVEL,08D
0374 E9A200              390                    JMP    SIZ_IS_SET
```

```
LOC  OBJ                 LINE     SOURCE

0377 3DF403               391     WAIT0:        CMP     ,1012D  ; WAS 1000
037A 7513                 392                   JNE     WAIT2
037C C70609000700    R    393                   MOV     SIZ,7D
0382 C606070007      R    394                   MOV     GSCALE_NUM,7D
0387 C6063E0007      R    395                   MOV     POP_LEVEL,7
038C E98A00               396                   JMP     SIZ_IS_SET
038F 3DEE05               397     WAIT2:        CMP     ,1518D  ;WAS 1500
0392 7513                 398                   JNE     WAIT3
0394 C70609000600    R    399                   MOV     SIZ,6D
039A C606070006      R    400                   MOV     GSCALE_NUM,6D
039F C6063E0006      R    401                   MOV     POP_LEVEL,6
03A4 EB7390               402                   JMP     SIZ_IS_SET
03A7 3DD207               403     WAIT3:        CMP     ,2002D  ;WAS 2000
03AA 7513                 404                   JNE     WAIT4
03AC C70609000500    R    405                   MOV     SIZ,5D
03B2 C606070005      R    406                   MOV     GSCALE_NUM,5D
03B7 C6063E0006      R    407                   MOV     POP_LEVEL,6
03BC EB5B90               408                   JMP     SIZ_IS_SET
03BF 3DC60B               409     WAIT4:        CMP     ,3014D ;WAS 3000
03C2 7513                 410                   JNE     WAIT7
03C4 C70609000400    R    411                   MOV     SIZ,4D
03CA C606070003      R    412                   MOV     GSCALE_NUM,3D
03CF C6063E0006      R    413                   MOV     POP_LEVEL,6
03D4 EB4390               414                   JMP     SIZ_IS_SET
03D7 3D9813               415     WAIT7:        CMP     ,5016D
03DA 7513                 416                   JNE     WAIT5
03DC C70609000400    R    417                   MOV     SIZ,4
03E2 C606070002      R    418                   MOV     GSCALE_NUM,2
03E7 C6063E0004      R    419                   MOV     POP_LEVEL,4
03EC EB2B90               420                   JMP     SIZ_IS_SET
03EF 3D6A1B               421     WAIT5:        CMP     ,7018D
03F2 7510                 422                   JNE     WAIT50
03F4 C70609000400    R    423                   MOV     SIZ,4D
03FA C606070001      R    424                   MOV     GSCALE_NUM,1
03FF C6063E0004      R    425                   MOV     POP_LEVEL,4
0404 3D3C23               426     WAIT50:       CMP     ,9020D
0407 7510                 427                   JNE     SIZ_IS_SET
0409 C70609000400    R    428                   MOV     SIZ,4D
040F C606070001      R    429                   MOV     GSCALE_NUM,1
0414 C6063E0002      R    430                   MOV     POP_LEVEL,2  ;THIS ENDS THE SIZ SETTING ROUTINE.
0419 C3                   431     SIZ_IS_SET:   RET
                          432     ;*********************************************************************
041A C606210000      R    433     SIZ_IS_SET2:  MOV     COUNT,0
                          434
041F E412                 435     STILL_SMOKE:  IN      AL,FLAGS  ;CHECK TO SEE IF MATROX IS STILL BUSY
0421 3C01                 436                   CMP     AL,1      ;1=NOT BUSY, 0=BUSY
0423 740C                 437                   JE      GO_ON2    ;JMP IF MATROX IS NOT BUSY
0425 26A00300             438                   MOV     AL,ES:DATA_READY  ;MATROX IS BUSY=>CHECK FOR DATA READY
0429 3C01                 439                   CMP     AL,1
042B 75F2                 440                   JNE     STILL_SMOKE
042D C3                   441                   RET               ;IF DATA IS READY WE BAIL OUT AND RET
042E EB1B90               442     DRAW_OCT_SHORT: JMP   DRAW_OCT
0431 803E210000      R    443     GO_ON2:       CMP     COUNT,0
0436 74F6                 444                   JE      DRAW_OCT_SHORT  ;DRAW NEW OCTAGON
0438 E821FF               445                   CALL    S_P_AND_GSCALE_SET ;SETS SIZ,POP DURING SMOKE GEN
```

```
LOC  OBJ                  LINE    SOURCE

043B E80000          E     446    SIZ_IS_SET3:     CALL    SMOKE_SET
043E A01F00          R     447                     MOV     AL,BACKGROUND
0441 E610                  448                     OUT     GSCALE,AL
0443 E414                  449                     IN      AL,ERASE  ;ERASE SCREEN
0445 C606210000      R     450                     MOV     COUNT,0
044A C3                    451                     RET
                           452
044B 803E210000      R     453    DRAW_OCT:        CMP     COUNT,0  ;ARE WE DRAWING OR ERASING? COUNT=0=DRAW.
0450 752F                  454                     JNE     BY_PAS
0452 803E080001      R     455                     CMP     THRUSTER_FIRE,1 ;AND IS THERE A THRUSTER FIRE??
0457 7508                  456                     JNE     CHK_COUNT_EM
0459 C6061E0003      R     457                     MOV     COUNT_EM,3
045E EB0890                458                     JMP     DO_IT_HERE
0461 803E1E0000      R     459    CHK_COUNT_EM:    CMP     COUNT_EM,0
0466 7419                  460                     JE      BY_PAS
0468 C606080000      R     461    DO_IT_HERE:      MOV     THRUSTER_FIRE,0 ;ZERO THIS FLAG . AS WE HAVE SEEN IT.
046D FE0E1E00        R     462                     DEC     COUNT_EM
0471 A00600          R     463                     MOV     AL,GSCALE_VAL
0474 02063E00        R     464                     ADD     AL,POP_LEVEL ;IF THERE IS THEN INC GSCALE_VAL BY P_L
0478 3C0F                  465                     CMP     AL,15D  ;HOWEVER,GSCALE_VAL CANNOT BE GREATER THAN 15D
047A 7602                  466                     JBE     OK
047C B00F                  467                     MOV     AL,15D  ;IF GREATER THAN 15 THEN LOWER TO 15
047E A20600          R     468    OK:              MOV     GSCALE_VAL,AL
0481 A00600          R     469    BY_PAS:          MOV     AL,GSCALE_VAL
0484 38061F00        R     470                     CMP     BACKGROUND,AL
0488 7206                  471                     JB      THERE
048A C606210001      R     472                     MOV     COUNT,1
048F C3                    473                     RET             ;NO NEED TO DRAW OCTAGON IF SMOKE WILL HIDE IT
                           474
0490 26A01400              475    THERE:           MOV     AL,ES:BIRD_DTA_RDY
0494 3C01                  476                     CMP     AL,1
0496 75F8                  477                     JNE     THERE
0498 26C606140000          478                     MOV     ES:BIRD_DTA_RDY,0
049E E80600                479                     CALL    OCT_DRAW
04A1 C606210001      R     480                     MOV     COUNT,1
04A6 C3                    481    FIN_OF_MATROX:   RET
                           482
                           483
04A7 833E390000      R     484    OCT_DRAW:        CMP     YANG2,0
04AC 7509                  485                     JNE     NEXT
04AE C7062D000000    R     486                     MOV     YANG_SCALED,0
04B4 EB0C90                487                     JMP     NEXT1
04B7 A13900          R     488    NEXT:            MOV     AX,YANG2
04BA 99                    489                     CWD
04BB F73E2B00        R     490                     IDIV    TWO
04BF A32D00          R     491                     MOV     YANG_SCALED,AX
04C2 833E3B0000      R     492    NEXT1:           CMP     ZANG2,0
04C7 7509                  493                     JNE     NEXT2
04C9 C7062F000000    R     494                     MOV     ZANG_SCALED,0
04CF EB0C90                495                     JMP     NEXT3
04D2 A13B00          R     496    NEXT2:           MOV     AX,ZANG2
04D5 99                    497                     CWD
04D6 F73E2B00        R     498                     IDIV    TWO
04DA A32F00          R     499                     MOV     ZANG_SCALED,AX
04DD A10900          R     500    NEXT3:           MOV     AX,SIZ  ;THIS STARTS THE OCTAGON DRAW ROUTINE!!!
```

```
LOC  OBJ                LINE     SOURCE

04E0 D01E8               501                  SHR     AX,1
04E2 BA7F00              502                  MOV     DX,127D ;THIS WAS XCNT
04E5 03162D00       R    503                  ADD     DX,YANG_SCALED
04E9 B97F00              504                  MOV     CX,127D ;THIS WAS YCNT
04EC 030E2F00       R    505                  ADD     CX,ZANG_SCALED
04F0 2BD0                506                  SUB     DX,AX
04F2 2BC8                507                  SUB     CX,AX
04F4 89160B00       R    508                  MOV     XMIN,DX
04F8 890E0D00       R    509                  MOV     YMIN,CX
04FC A10900         R    510                  MOV     AX,SIZ
04FF 03C2                511                  ADD     AX,DX
0501 A30F00         R    512                  MOV     XMAX,AX
0504 A10900         R    513                  MOV     AX,SIZ
0507 03C1                514                  ADD     AX,CX
0509 A31100         R    515                  MOV     YMAX,AX
050C A10900         R    516                  MOV     AX,SIZ
050F F6361D00       R    517                  DIV     THREE
0513 80FC01              518                  CMP     AH,1
0516 7602                519                  JBE     GO_ON
0518 FEC0                520                  INC     AL
051A 8AD8                521     GO_ON:       MOV     BL,AL
051C 32E4                522                  XOR     AH,AH
051E 03060D00       R    523                  ADD     AX,YMIN
0522 A31900         R    524                  MOV     ONE_THIRD,AX
0525 8AC3                525                  MOV     AL,BL
0527 D0E0                526                  SHL     AL,1
0529 32E4                527                  XOR     AH,AH
052B 03060D00       R    528                  ADD     AX,YMIN
052F A31B00         R    529                  MOV     TWO_THIRDS,AX
0532 A11900         R    530     FIRST_THIRD: MOV     AX,ONE_THIRD
0535 2BC1                531                  SUB     AX,CX
0537 A31700         R    532                  MOV     TMP_WRD,AX
053A 03060B00       R    533                  ADD     AX,XMIN
053E A31500         R    534                  MOV     XMIN_TMP,AX ;SETS XMIN
0541 A10F00         R    535                  MOV     AX,XMAX
0544 2B061700       R    536                  SUB     AX,TMP_WRD
0548 A31300         R    537                  MOV     XMAX_TMP,AX  ;SETS XMAX
054E 803E3D0001     R    538                  CMP     FLAG,1
0550 740D                539                  JE      OVER_THIS
0552 833E090005     R    540                  CMP     SIZ,5
0557 7506                541                  JNE     OVER_THIS
0559 41                  542                  INC     CX
055A C6063D0001     R    543                  MOV     FLAG,1
055F E86600              544     OVER_THIS:   CALL    LINE
0562 41                  545                  INC     CX
0563 3B0E1900       R    546                  CMP     CX,ONE_THIRD
0567 7EC9                547                  JLE     FIRST_THIRD
0569 C6063D0000     R    548                  MOV     FLAG,0
056E A10B00         R    549     SECOND_THIRD MOV     AX,XMIN
0571 A31500         R    550                  MOV     XMIN_TMP,AX
0574 A10F00         R    551                  MOV     AX,XMAX
0577 A31300         R    552                  MOV     XMAX_TMP,AX
057A E84B00              553     OVER1:       CALL    LINE
057D 41                  554                  INC     CX
057E 3B0E1B00       R    555                  CMP     CX,TWO_THIRDS
```

```
LOC  OBJ                  LINE     SOURCE

                           556                    JLE     OVER1
0582 7EF6                  557     LAST_THIRD:    MOV     AX,TWO_THIRDS
0584 A11B00          R     558                    MOV     BX,CX
0587 8BD9                  559                    SUB     BX,AX
0589 2BD8                  560                    MOV     AX,BX
058B 8BC3                  561                    MOV     TMP_WRD,AX
058D A31700          R     562                    ADD     AX,XMIN
0590 03060B00        R     563                    MOV     XMIN_TMP,AX
0594 A31500          R     564                    MOV     AX,XMAX
0597 A10F00          R     565                    SUB     AX,TMP_WRD
059A 2B061700        R     566                    MOV     XMAX_TMP,AX
059E A31300          R     567                    CMP     FLAG,1
05A1 803E3D0001      R     568                    JE      OVER_THIS_2
05A6 7410                  569                    CMP     SIZ,4
05A8 833E090004      R     570                    JNE     OVER_THIS_2
05AD 7509                  571                    DEC     YMAX
05AF FF0E1100        R     572                    MOV     FLAG,1
05B3 C6063D0001      R     573     OVER_THIS_2:   CALL    LINE
05B8 E80000                574                    INC     CX
05BB 41                    575                    CMP     CX,YMAX
05BC 3B0E1100        R     576                    JLE     LAST_THIRD
05C0 7EC2                  577                    MOV     FLAG,0
05C2 C6063D0000      R     578                    RET
05C7 C3                    579
                           580
                           581     ;THE ROUTINE DOT WILL NOT BE USED FOR A WHILE
                           582     ;              MOV     AL,SIZ
                           583     ;              CMP     AL,3
                           584     ;              JE      DOT
                           585     ;              SUB     SIZ,3D
                           586     ;              JMP     START
                           587     ;DOT:           MOV     DL,ES:X
                           588     ;              MOV     CL,ES:Y
                           589     ;              MOV     XMAX_TMP,DL
                           590     ;              CALL    OVER
                           591     ;              MOV     AL,BACKGROUND
                           592     ;              OUT     GSCALE,AL
                           593     ;              IN      AL,ERASE
                           594     ;              HLT
                           595
                           596
05C8 8B161500        R     597     LINE:          MOV     DX,XMIN_TMP
05CC 83F900                598                    CMP     CX,0D
05CF 7E20                  599                    JLE     DUN
05D1 81F9FF00              600                    CMP     CX,255D
05D5 7D1A                  601                    JGE     DUN
05D7 BBFF00                602                    MOV     BX,255D
05DA 2BD9                  603                    SUB     BX,CX
05DC 81FAFF00              604     OVER:          CMP     DX,255D
05E0 7D08                  605                    JGE     NO_WAY
05E2 83FA00                606                    CMP     DX,0D
05E5 7E03                  607                    JLE     NO_WAY
05E7 E80000                608                    CALL    OUTPUT
05EA 42                    609     NO_WAY:        INC     DX
05EB 3B161300        R     610                    CMP     DX,XMAX_TMP
```

```
LOC  OBJ                  LINE    SOURCE

05EF 7EEB                  611                       JLE     OVER
05F1 C3                    612    DUN:               RET
                           613
05F2 8AC2                  614    OUTPUT:            MOV     AL,DL
05F4 E614                  615                       OUT     XREG,AL
05F6 8AC3                  616                       MOV     AL,BL
05F8 E616                  617                       OUT     YREG,AL
05FA A00600          R     618                       MOV     AL,GSCALE_VAL
05FD E610                  619                       OUT     GSCALE,AL
05FF C3                    620                       RET
                           621
0600 26C606170000          622    PRINT_RESULTS:     MOV     ES:BIRD_MISSES,0
0606 C606290001      R     623                       MOV     RESULTS_FLAG,1
060B BAD800                624                       MOV     DX,0D8H
060E B01D                  625                       MOV     AL,0350 ;VECTOR MODE
0610 E80000          E     626                       CALL    COUT
0613 B037                  627                       MOV     AL,0670
0615 E80000          E     628                       CALL    COUT
0618 B078                  629                       MOV     AL,1700
061A E80000          E     630                       CALL    COUT
061D B028                  631                       MOV     AL,0500
061F E80000          E     632                       CALL    COUT
0622 B040                  633                       MOV     AL,1000
0624 E80000          E     634                       CALL    COUT
0627 B01F                  635                       MOV     AL,0370 ;ALPHA MODE
0629 E80000          E     636                       CALL    COUT
062C B92000                637                       MOV     CX,32D
062F BE2000                638                       MOV     SI,OFFSET H_MIS_ASCII
0632 268A04                639    AGAIN:             MOV     AL,ES:[SI]
0635 E80000          E     640                       CALL    COUT
0638 E81502                641                       CALL    SHORT_DELAY
063B 46                    642                       INC     SI
063C E2F4                  643                       LOOP    AGAIN
063E B01D                  644                       MOV     AL,0350
0640 E80000          E     645                       CALL    COUT
0643 E9CAFA                646                       JMP     DO_IT_OVER
                           647
0646 26C6061E0000          648    DIST_FRO_TGT:      MOV     ES:HIT_SHORT,0
064C C606280001      R     649                       MOV     DIST_FRO_TGT_FLAG,1
0651 BAD800                650                       MOV     DX,0D8H
0654 B01D                  651                       MOV     AL,0350 ;VECTOR MODE
0656 E80000          E     652                       CALL    COUT
0659 B037                  653                       MOV     AL,0670
065B E80000          E     654                       CALL    COUT
065E B078                  655                       MOV     AL,1700
0660 E80000          E     656                       CALL    COUT
0663 B028                  657                       MOV     AL,0500
0665 E80000          E     658                       CALL    COUT
0668 B040                  659                       MOV     AL,1000
066A E80000          E     660                       CALL    COUT
066D B01F                  661                       MOV     AL,0370 ;ALPHA MODE
066F E80000          E     662                       CALL    COUT
0672 B91000                663                       MOV     CX,16D
0675 BE4000                664                       MOV     SI,OFFSET DIS_FRO_TGT
0678 268A04                665    AGN:               MOV     AL,ES:[SI]
```

```
LOC  OBJ               LINE    SOURCE

067B E80000        E    666                  CALL   COUT
067E E8CF01             667                  CALL   SHORT_DELAY
0681 46                 668                  INC    SI
0682 E2F4               669                  LOOP   AGN
0684 B01D               670                  MOV    AL,35O
0686 E80000        E    671                  CALL   COUT
0689 E984FA             672                  JMP    DO_IT_OVER
                        673
068C BAD800             674   THRUSTERS_MSG: MOV    DX,0D8H
068F B01D               675                  MOV    AL,035O
0691 E80000        E    676                  CALL   COUT
0694 B020               677                  MOV    AL,40O
0696 E80000        E    678                  CALL   COUT
0699 B060               679                  MOV    AL,140O
069B E80000        E    680                  CALL   COUT
069E B020               681                  MOV    AL,40O
06A0 E80000        E    682                  CALL   COUT
06A3 B040               683                  MOV    AL,100O
06A5 E80000        E    684                  CALL   COUT
06A8 B01F               685                  MOV    AL,37O
06AA E80000        E    686                  CALL   COUT
06AD B90B00             687                  MOV    CX,(OFFSET END_OF_MESSAGE3 - OFFSET MESSAGE3)
06B0 BE3500        R    688                  MOV    SI,OFFSET MESSAGE3
06B3 2E8A04             689   AGN7:          MOV    AL,CS:[SI]
06B6 E80000        E    690                  CALL   COUT
06B9 E89401             691                  CALL   SHORT_DELAY
06BC 46                 692                  INC    SI
06BD E2F4               693                  LOOP   AGN7
06BF B90200             694                  MOV    CX,2
06C2 BE6B00             695                  MOV    SI,OFFSET (IDEAL_THRUST +3)
06C5 268A04             696   IT_AGAIN:      MOV    AL,ES:[SI]
06C8 E80000        E    697                  CALL   COUT
06CB E86201             698                  CALL   SHORT_DELAY
06CE 46                 699                  INC    SI
06CF E2F4               700                  LOOP   IT_AGAIN
06D1 B049               701                  MOV    AL,111O
06D3 E80000        E    702                  CALL   COUT
06D6 B02F               703                  MOV    AL,57O
06D8 E80000        E    704                  CALL   COUT
06DB B90200             705                  MOV    CX,2
06DE BE5300             706                  MOV    SI,OFFSET (ACTUAL_THRUST+3)
06E1 268A04             707   IT_AGAIN2:     MOV    AL,ES:[SI]
06E4 E80000        E    708                  CALL   COUT
06E7 E86601             709                  CALL   SHORT_DELAY
06EA 46                 710                  INC    SI
06EB E2F4               711                  LOOP   IT_AGAIN2
06ED B041               712                  MOV    AL,101O
06EF E80000        E    713                  CALL   COUT
06F2 B01D               714                  MOV    AL,35O
06F4 E80000        E    715                  CALL   COUT
06F7 C3                 716                  RET
                        717
                        718
06F8 E891FF             719   PRINT_HIT:     CALL   THRUSTERS_MSG
06FB C606220003    R    720                  MOV    BLINK_COUNT,03D
```

```
LOC  OBJ                  LINE     SOURCE

0700 C606270001      R     721                     MOV     HIT_FLAG,1
0705 26C606160000          722                     MOV     ES:BIRD_HIT,0
0708 BAD800                723     PRINT_HIT_AGN:  MOV     DX,0D8H
070E B01D                  724                     MOV     AL,0350 ;VECTOR MODE
0710 E80000          E     725                     CALL    COUT
0713 B037                  726                     MOV     AL,0670
0715 E80000          E     727                     CALL    COUT
0718 B076                  728                     MOV     AL,1700
071A E80000          E     729                     CALL    COUT
071D B02E                  730                     MOV     AL,0560
071F E80000          E     731                     CALL    COUT
0722 B042                  732                     MOV     AL,1020
0724 E80000          E     733                     CALL    COUT
0727 B01F                  734                     MOV     AL,0370 ;ALPHA MODE
0729 E80000          E     735                     CALL    COUT
072C B048                  736                     MOV     AL,1100 ;H
072E E80000          E     737                     CALL    COUT
0731 B049                  738                     MOV     AL,1110 ;I
0733 E80000          E     739                     CALL    COUT
0736 B054                  740                     MOV     AL,1240 ;T
0738 E80000          E     741                     CALL    COUT
073B B021                  742                     MOV     AL,0410 ;!
073D E80000          E     743                     CALL    COUT
0740 B01D                  744                     MOV     AL,0350
0742 E80000          E     745                     CALL    COUT
0745 E8FE00                746                     CALL    DELAY
0748 B01D                  747                     MOV     AL,0350
074A E80000          E     748                     CALL    COUT
074D 26803E1A0001          749                     CMP     ES:V_REP_REQ,1 ;ARE "WE"NEEDED ANYWHERE??
0753 7464                  750                     JE      DOWN
0755 26803E180001          751                     CMP     ES:H_REP_REQ,1 ;ARE "WE" NEEDED ANYWHERE??
075B 745C                  752                     JE      DOWN
075D FE0E2200        R     753                     DEC     BLINK_COUNT
0761 803E220000      R     754                     CMP     BLINK_COUNT,0
0766 7451                  755                     JE      DOWN
0768 B01B                  756                     MOV     AL,0330 ;
076A E80000          E     757                     CALL    COUT    ;
076D B07F                  758                     MOV     AL,1770 ;THESE FOUR INSTRUCTIONS CHANGE THE
076F E80000          E     759                     CALL    COUT    ;DATA LEVEL TO BLACK!
0772 B037                  760                     MOV     AL,670
0774 E80000          E     761                     CALL    COUT
0777 B076                  762                     MOV     AL,1700
0779 E80000          E     763                     CALL    COUT
077C B02E                  764                     MOV     AL,560
077E E80000          E     765                     CALL    COUT
0781 B042                  766                     MOV     AL,1020
0783 E80000          E     767                     CALL    COUT
0786 B01F                  768                     MOV     AL,370
0788 E80000          E     769                     CALL    COUT ;ALPHA MODE
078B B048                  770                     MOV     AL,1100 ;"BLACK 'H'"
078D E80000          E     771                     CALL    COUT
0790 B049                  772                     MOV     AL,1110 ;"BLACK 'I'"
0792 E80000          E     773                     CALL    COUT
0795 B054                  774                     MOV     AL,1240 ;"BLACK 'T'"
0797 E80000          E     775                     CALL    COUT
```

```
LOC  OBJ                 LINE     SOURCE

079A B021                 776                     MOV    AL,410 ;"BLACK '!'"
079C E80000         E     777                     CALL   COUT
079F B021                 778                     MOV    AL,410
07A1 E80000         E     779                     CALL   COUT
07A4 B01B                 780                     MOV    AL,330 ;THESE NEXT FOUR INSTRUCTIONS SET
07A6 E80000         E     781                     CALL   COUT
07A9 B061                 782                     MOV    AL,1410 ;THE DATA LEVEL BACK TO WHITE
07AB E80000         E     783                     CALL   COUT
07AE B01D                 784                     MOV    AL,350 ;RETURN TO VECTOR MODE
07B0 E80000         E     785                     CALL   COUT
07B3 E89A00               786                     CALL   SHORT_DELAY
07B6 E952FF               787                     JMP    PRINT_HIT_AGN
07B9 E954F9               788     DOWN:           JMP    DO_IT_OVER
                          789
07BC E8CDFE               790     PRINT_GRND_BIRD: CALL  THRUSTERS_MSG
07BF 26C6061C0000         791                     MOV    ES:GRND_BIRD,0
07C5 C606260001     R     792                     MOV    GRND_BIRD_FLAG,1
07CA BAD800               793                     MOV    DX,0D8H
07CD B01D                 794                     MOV    AL,0350
07CF E80000         E     795                     CALL   COUT
07D2 B037                 796                     MOV    AL,0670
07D4 E80000         E     797                     CALL   COUT
07D7 B061                 798                     MOV    AL,1410
07D9 E80000         E     799                     CALL   COUT
07DC B02E                 800                     MOV    AL,0560
07DE E80000         E     801                     CALL   COUT
07E1 B042                 802                     MOV    AL,1020
07E3 E80000         E     803                     CALL   COUT
07E6 B01F                 804                     MOV    AL,0370 ;ALPHA MODE
07E8 E80000         E     805                     CALL   COUT
07EB B90E00               806                     MOV    CX,(OFFSET FIN_OF_MESSAGE - OFFSET MESSAGE)
07EE BE0000         R     807                     MOV    SI,OFFSET MESSAGE
07F1 2E8A04               808     REPEAT:         MOV    AL,CS:[SI]
07F4 E80000         E     809                     CALL   COUT
07F7 E85600               810                     CALL   SHORT_DELAY
07FA 46                   811                     INC    SI
07FB E2F4                 812                     LOOP   REPEAT
07FD E910F9               813                     JMP    DO_IT_OVER
                          814
0800 C606240000     R     815     ERROR_MESG:     MOV    ERROR,0
0805 C606250001     R     816                     MOV    ERROR_MESG_FLAG,1
080A 26C6061D0000         817                     MOV    ES:END_OF_RPV,0
0810 BAD800               818                     MOV    DX,0D8H
0813 B01D                 819                     MOV    AL,350
0815 E80000         E     820                     CALL   COUT
0818 B020                 821                     MOV    AL,400
081A E80000         E     822                     CALL   COUT
081D B060                 823                     MOV    AL,1400
081F E80000         E     824                     CALL   COUT
0822 B02E                 825                     MOV    AL,560
0824 E80000         E     826                     CALL   COUT
0827 B041                 827                     MOV    AL,1010
0829 E80000         E     828                     CALL   COUT
082C B01F                 829                     MOV    AL,370 ;ALPHA MODE
082E E80000         E     830                     CALL   COUT
```

```
LOC  OBJ                 LINE    SOURCE

0831 B92700               831                    MOV   CX,(OFFSET FIN_OF_MESSAGE2 - OFFSET MESSAGE2)
0834 BE0E00        R      832                    MOV   SI,OFFSET MESSAGE2
0837 2E8A04               833    REPT2:          MOV   AL,CS:[SI]
083A E80000        E      834                    CALL  COUT
083D E81000               835                    CALL  SHORT_DELAY
0840 46                   836                    INC   SI
0841 E2F4                 837                    LOOP  REPT2
0843 E9CAF8               838                    JMP   DO_IT_OVER
                          839
0846 B850C3               840    DELAY:          MOV   AX,50000D
0849 48                   841    AGAIN_N_AGAIN:  DEC   AX
084A 3D0000               842                    CMP   AX,0
084D 75FA                 843                    JNE   AGAIN_N_AGAIN
084F C3                   844                    RET
0850 B81027               845    SHORT_DELAY:    MOV   AX,10000D
0853 48                   846    AGN_N_AGN:      DEC   AX
0854 3D0000               847                    CMP   AX,0
0857 75FA                 848                    JNE   AGN_N_AGN
0859 C3                   849                    RET
                          850
085A B400                 851    CHECK_FOR_B:    MOV   AH,0
085C E40E                 852                    IN    AL,0EH    ;USART_STATUS
085E 2402                 853                    AND   AL,2
0860 7404                 854                    JZ    GO_BACK
0862 E40C                 855                    IN    AL,0CH
0864 B401                 856                    MOV   AH,1
0866 C3                   857    GO_BACK:        RET
                          858
0867 268B160100           859    CALIBRATE:      MOV   DX,ES:X_Y
086C B064                 860                    MOV   AL,64H
086E 2AC2                 861                    SUB   AL,DL
0870 2600060500           862                    ADD   ES:OFFSET_X,AL
0875 B064                 863                    MOV   AL,64H
0877 2AC6                 864                    SUB   AL,DH
0879 2600060600           865                    ADD   ES:OFFSET_Y,AL
087E C3                   866                    RET
                          867
----                      868    CODE            ENDS
                          869
                          870                    END   CS:DRIVER,DS:DATA_GROUP,SS:STACK_SEG

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

```
SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.0 ASSEMBLY OF MODULE SMOKEY_THE_BARE
OBJECT MODULE PLACED IN :F2:SMOKE.OBJ
INVOCATION LINE CONTROLS:  DEBUG


LOC  OBJ                  LINE     SOURCE

                            1      NAME    SMOKEY_THE_BARE
                            2
                            3      DATA_GROUP       GROUP   DAT_SEG,DATA_SEG
                            4      CGROUP           GROUP   CODE
                            5
                            6      ASSUME CS:CGROUP,DS:DATA_GROUP
                            7
                            8      PUBLIC  SMOKE_CHECK,SMOKE_SET,SMOKE_START_UP
                            9
----                       10      DATA_SEG         SEGMENT PUBLIC
                           11      EXTRN   BACKGROUND: BYTE
----                       12      DATA_SEG         ENDS
                           13
----                       14      DAT_SEG SEGMENT PUBLIC
0000 (1                    15      DONE             DB      1 DUP(?)
     ??
     )
0001 (1                    16      SERIES_NO        DB      1 DUP(?)
     ??
     )
----                       17      DAT_SEG ENDS
                           18
----                       19      CODE             SEGMENT PUBLIC  'CODE'
0000 C606000000     R      20      SMOKE_START_UP: MOV     DONE,0
0005 C606010001     R      21                       MOV     SERIES_NO,1
000A C3                    22                       RET
                           23
000B 803E000000     R      24      SMOKE_CHECK:     CMP     DONE,0
0010 7405                  25                       JE      GO_BACK
0012 B401                  26                       MOV     AH,1
0014 EB0390                27                       JMP     GO_BACK2
0017 B400                  28      GO_BACK:         MOV     AH,0
0019 C3                    29      GO_BACK2:        RET
                           30
001A 803E000000     R      31      SMOKE_SET:       CMP     DONE,0
001F 7403                  32                       JE      START
0021 B401                  33                       MOV     AH,1
0023 C3                    34                       RET
                           35
                           36
0024 803E010001     R      37      START:           CMP     SERIES_NO,1
0029 741F                  38                       JE      SERIES_1
002B 803E010002     R      39                       CMP     SERIES_NO,2
0030 742A                  40                       JE      SERIES_2
0032 803E010003     R      41                       CMP     SERIES_NO,3
0037 7435                  42                       JE      SERIES_3
0039 803E010004     R      43                       CMP     SERIES_NO,4
003E 7440                  44                       JE      SERIES_4
0040 803E010005     R      45                       CMP     SERIES_NO,5
0045 744B                  46                       JE      SERIES_5
```

```
LOC  OBJ                  LINE     SOURCE

0047 EB5B90                 47                     JMP     SERIES_6
                            48
004A FE060000        E      49     SERIES_1:       INC     BACKGROUND
004E 803E00000F      E      50                     CMP     BACKGROUND,15D
0053 7401                   51                     JE      DONE1
0055 C3                     52                     RET
0056 C606010002      R      53     DONE1:          MOV     SERIES_NO,2
005B C3                     54                     RET
                            55
005C FE0E0000        E      56     SERIES_2:       DEC     BACKGROUND
0060 803E000008      E      57                     CMP     BACKGROUND,8D
0065 7401                   58                     JE      DONE2
0067 C3                     59                     RET
0068 C606010003      R      60     DONE2:          MOV     SERIES_NO,3
006D C3                     61                     RET
                            62
006E FE060000        E      63     SERIES_3:       INC     BACKGROUND
0072 803E00000F      E      64                     CMP     BACKGROUND,15D
0077 7401                   65                     JE      DONE3
0079 C3                     66                     RET
007A C606010004      R      67     DONE3:          MOV     SERIES_NO,4
007F C3                     68                     RET
                            69
0080 FE0E0000        E      70     SERIES_4:       DEC     BACKGROUND
0084 803E000004      E      71                     CMP     BACKGROUND,4
0089 7401                   72                     JE      DONE4
008B C3                     73                     RET
008C C606010005      R      74     DONE4:          MOV     SERIES_NO,5
0091 C3                     75                     RET
                            76
0092 FE060000        E      77     SERIES_5        INC     BACKGROUND
0096 803E000008      E      78                     CMP     BACKGROUND,8D
009B 7401                   79                     JE      DONE5
009D C3                     80                     RET
009E C606010006      R      81     DONE5:          MOV     SERIES_NO,6
00A3 C3                     82                     RET
                            83
00A4 FE0E0000        E      84     SERIES_6:       DEC     BACKGROUND
00A8 803E000000      E      85                     CMP     BACKGROUND,0
00AD 7401                   86                     JE      FINISHED
00AF C3                     87                     RET
00B0 C606000001      R      88     FINISHED:       MOV     DONE,1
00B5 C3                     89                     RET
                            90
----                        91     CODE            ENDS
                            92                     END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

```
SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.0 ASSEMBLY OF MODULE PETER_RABBIT
OBJECT MODULE PLACED IN :F2:GAE.OBJ
INVOCATION LINE CONTROLS: DEBUG


LOC  OBJ                  LINE     SOURCE

                             1     NAME    PETER_RABBIT
                             2
                             3     DATA_GROUP      GROUP   GAE_SEG,DATA_SEG,XFER_SEG
                             4     CGROUP          GROUP   CODE
                             5
                             6     EXTRN GRAPH1:NEAR,FIRE_BIRD:BYTE
                             7     PUBLIC GRAPH_GAE_POINT,GAE_START_UP,GAE_GT_FOV,H_REP_FLAG,V_REP_FLAG
                             8     PUBLIC ELAPSED_TIME
                             9     EXTRN COUT:NEAR
                            10
                            11     ASSUME CS:CGROUP,DS:DATA_GROUP
                            12
----                        13     XFER_SEG        SEGMENT AT 0600H
                            14     EXTRN BAD_MISS:BYTE,START_BIT:BYTE,DATA_READY:BYTE
----                        15     XFER_SEG ENDS
                            16
----                        17     DATA_SEG        SEGMENT PUBLIC
                            18     EXTRN YCNT:BYTE,XCNT:BYTE,THRUSTER_FIRE:BYTE,ERROR:BYTE
----                        19     DATA_SEG ENDS
                            20
----                        21     GAE_SEG SEGMENT PUBLIC
                            22
0000 (1                     23     BIT_BUCKET              DB      1 DUP(?)
      ??
      )
0001 (1                     24     ELAPSED_TIME            DW      1 DUP(?)
      ????
      )
0003 (1                     25     FIFTEEN                 DW      1 DUP(?)
      ????
      )
0005 (1                     26     FOUR                    DW      1 DUP(?)
      ????
      )
  0003                      27     V_SCALE_FACTOR          EQU     03D
0007 (1                     28     H_X_GRAPHIC_POINT       DW      1 DUP(?)
      ????
      )
0009 (1                     29     H_Y_GRAPHIC_POINT       DW      1 DUP(?)
      ????
      )
000B (1                     30     HGAE_OFFSET             DW      1 DUP(?)
      ????
      )
000D (1                     31     FIFTY                   DW      1 DUP(?)
      ????
      )
000F (1                     32     H_REP_FLAG              DB      1 DUP(?)
      ??
      )
```

```
LOC  OBJ                  LINE     SOURCE

0010 (1                     33     V_REP_FLAG              DB       1 DUP(?)
     ??
     )

0011 (1                     34     BIG_MISS                DB       1 DUP(?)
     ??
     )

0012 (1                     35     THRUSTER_FIRED          DB       1 DUP(?)
     ??
     )
                            36
----                        37     GAE_SEG         ENDS
                            38
----                        39     CODE    SEGMENT PUBLIC 'CODE'
                            40
0000 C606000000       R     41     GAE_START_UP:   MOV     BIT_BUCKET,0
0005 C606110000       R     42                     MOV     BIG_MISS,0
000A C70601000000     R     43                     MOV     ELAPSED_TIME,0
0010 C70603000F00     R     44                     MOV     FIFTEEN,15D
0016 C70605000400     R     45                     MOV     FOUR,4D
001C C7060D003200     R     46                     MOV     FIFTY,50D
0022 C7060B00D502     R     47                     MOV     HGAE_OFFSET,725D
0028 C606120000       R     48                     MOV     THRUSTER_FIRED,0
002D C6060F0000       R     49                     MOV     H_REP_FLAG,0
0032 C606100000       R     50                     MOV     V_REP_FLAG,0
0037 C3                     51                     RET
                            52
0038 BAD800                 53     GAE_GT_FOV:     MOV     DX,0D8H
003B 26C606000000     E     54                     MOV     ES:DATA_READY,0
0041 B01D                   55                     MOV     AL,035O ;VECTOR MODE
0043 E80000           E     56                     CALL    COUT
0046 E80900                 57                     CALL    CONTINUE
0049 C3                     58                     RET
                            59
004A BAD800                 60     GRAPH_GAE_POINT:        MOV     DX,0D8H
004D B01C                   61                             MOV     AL,034O
004F E80000           E     62                             CALL    COUT
0052 33C0                   63     CONTINUE:               XOR     AX,AX
0054 803E000001       R     64                             CMP     BIT_BUCKET,1 ;IF SET THEN WE'RE ON PASS TWO
0059 740D                   65                             JE      NORMAL        ;I.E.,LEFT SIDE OF SCREEN
005B 803E100001       R     66                             CMP     V_REP_FLAG,1
0060 7506                   67                             JNE     NORMAL
0062 C7060B004D03     R     68                             MOV     HGAE_OFFSET,845D
0068 8B1E0B00         R     69     NORMAL:                 MOV     BX,HGAE_OFFSET
006C 81FBD502               70                             CMP     BX,725D
0070 7412                   71                             JZ      X1
0072 81FB4D03               72                             CMP     BX,845D
0076 740C                   73                             JZ      X1
0078 A00000           E     74                             MOV     AL,YCNT
007B B3C8                   75                             MOV     BL,200D
007D 2AD8                   76                             SUB     BL,AL
007F 8AC3                   77                             MOV     AL,BL
0081 EB0490                 78                             JMP     ONWARD
0084 A00000           E     79     X1:                     MOV     AL,XCNT
0087 B400                   80     ONWARD:                 MOV     AH,0
0089 F7260300         R     81                             MUL     FIFTEEN
```

```
LOC  OBJ                 LINE    SOURCE

008D F7360500        R     82                          DIV    FOUR
0091 83FA02                83                          CMP    DX,02D
0094 7C01                  84                          JL     GO_ON
0096 40                    85                          INC    AX
0097 8B1E0B00        R     86    GO_ON:                MOV    BX,HGAE_OFFSET
009B B97701                87                          MOV    CX,375D
009E 81FBD502              88                          CMP    BX,725D
00A2 7413                  89                          JZ     X2
00A4 81FB4D03              90                          CMP    BX,845D
00A8 740D                  91                          JZ     X2
00AA 8A1E0000        E     92                          MOV    BL,YCNT
00AE B2C8                  93                          MOV    DL,200D
00B0 2AD3                  94                          SUB    DL,BL
00B2 8ADA                  95                          MOV    BL,DL
00B4 EB0590                96                          JMP    ONWARD1
00B7 8A1E0000        E     97    X2:                   MOV    BL,XCNT
00BB 80FB64                98    ONWARD1:              CMP    BL,100D
00BE 7314                  99                          JNB    SUBT
00C0 2BC8                 100    AD:                   SUB    CX,AX
00C2 8BC1                 101                          MOV    AX,CX
00C4 8B1E0B00        R    102                          MOV    BX,HGAE_OFFSET
00C8 03060B00        R    103                          ADD    AX,HGAE_OFFSET
00CC 3DF203               104                          CMP    AX,1010D
00CF 7F15                 105                          JG     ONWARD3
00D1 EB0D90               106                          JMP    GO_ON5
00D4 2BC1                 107    SUBT:                 SUB    AX,CX
00D6 8B1E0B00        R    108                          MOV    BX,HGAE_OFFSET
00DA 2BD8                 109                          SUB    BX,AX
00DC 7E11                 110                          JLE    ONWARD2
00DE 8BC3                 111                          MOV    AX,BX
00E0 A30700          R    112    GO_ON5:               MOV    H_X_GRAPHIC_POINT,AX
00E3 EB1090               113                          JMP    OVER_THERE
00E6 C7060700F203    R    114    ONWARD3:              MOV    H_X_GRAPHIC_POINT,1010D
00EC EB0790               115                          JMP    OVER_THERE
00EF C70607000000    R    116    ONWARD2:              MOV    H_X_GRAPHIC_POINT,0
00F5 E82B01               117    OVER_THERE:           CALL   TIME
00F8 BB0300               118                          MOV    BX,Y_SCALE_FACTOR
00FB F7E3                 119                          MUL    BX
00FD 05B80B               120                          ADD    AX,3000D
0100 F7360D00        R    121                          DIV    FIFTY
0104 83FA19               122                          CMP    DX,25D
0107 7C01                 123                          JL     GO_ON4
0109 40                   124                          INC    AX
                          125
010A 3D0B03               126    GO_ON4:               CMP    AX,779D  ;IS TIME GOING TO EXCEED THE TOP OF
010D 7E03                 127                          JNG    GO_ON6   ;THE ADM-3 SCREEN?? IF YES THEN REDUCE
010F B80B03               128                          MOV    AX,779D  ;THE MAGNITUDE TO 779D.
                          129
0112 A30900          R    130    GO_ON6:               MOV    H_Y_GRAPHIC_POINT,AX
0115 BAD000               131                          MOV    DX,0D0H
                          132    ;THIS ROUTINE OUTPUTS HIGH Y, LOW Y, HIIGH X AND LOW X FOR HORIZONTAL
                          133    ;AIM ERROR TO ADM3.
0118 26803E000001    E    134                          CMP    ES:BAD_MISS,1  ;BAD_MISS???
011E 7505                 135                          JNE    BY_PASS  ;IF YES THEN
0120 B01D                 136                          MOV    AL,035D  ;GO TO VECTOR MODE
```

```
LOC  OBJ                LINE    SOURCE

0122 E80000        E     137                                 CALL    COUT      ;ELSE,STAY IN POINT MODE
                         138
0125 A10900        R     129     BY_PASS:                    MOV     AX,H_Y_GRAPHIC_POINT
0128 B105                140                                 MOV     CL,5
012A D3E8                141                                 SHR     AX,CL
012C 241F                142                                 AND     AL,00011111B
012E 0C20                143                                 OR      AL,00100000B
0130 E80000        E     144                                 CALL    COUT
0133 A10900        R     145                                 MOV     AX,H_Y_GRAPHIC_POINT
0136 241F                146                                 AND     AL,00011111B
0138 0C60                147                                 OR      AL,01100000B
013A E80000        E     148                                 CALL    COUT
013D A10700        R     149                                 MOV     AX,H_X_GRAPHIC_POINT
0140 B105                150                                 MOV     CL,5
0142 D3E8                151                                 SHR     AX,CL
0144 241F                152                                 AND     AL,00011111B
0146 0C20                153                                 OR      AL,00100000B
0148 E80000        E     154                                 CALL    COUT
014B A10700        R     155                                 MOV     AX,H_X_GRAPHIC_POINT
014E 241F                156                                 AND     AL,00011111B
0150 0C40                157                                 OR      AL,01000000B
0152 E80000        E     158                                 CALL    COUT
0155 26803E000001  E     159                                 CMP     ES:BAD_MISS,1
015B 751E                160                                 JNE     BY_PASS2
015D B01F                161                                 MOV     AL,037O ;ALPHA MODE
015F E80000        E     162                                 CALL    COUT
0162 803E120001    R     163                                 CMP     THRUSTER_FIRED,1
0167 7508                164                                 JNE     AST
0169 B05F                165                                 MOV     AL,137O ;OUTPUT A '_' WHEN A THRUSTER FIRES.
016B E80000        E     166                                 CALL    COUT
016E EB0B90              167                                 JMP     BY_PASS2
0171 B02A                168     AST:                        MOV     AL,052O ;OUTPUT ASTERISK
0173 E80000        E     169                                 CALL    COUT
0176 B01D                170                                 MOV     AL,035O
0178 E80000        E     171                                 CALL    COUT
017B 803E000001    R     172     BY_PASS2:                   CMP     BIT_BUCKET,1
0180 743A                173                                 JE      AGAIN2
0182 832E010016    R     174                                 SUB     ELAPSED_TIME,22O  ;USED TO BE 20
0187 C606000001    R     175                                 MOV     BIT_BUCKET,1
018C 803E0F0001    R     176                                 CMP     H_REP_FLAG,1  ;IF REP THEN A DIFFERENT OFFSET
0191 7409                177                                 JE      ABNORMAL        ;FROM REGULAR IS USED
0193 C7060B00C300  R     178                                 MOV     HGRE_OFFSET,195D   ;CHANGE THE OFFSETS TO GET
0199 EB0790              179                                 JMP     ACROSS             ;READY TO DO THE VERTICALGRE
019C C7060B00EF00  R     180     ABNORMAL:                   MOV     HGRE_OFFSET,239D
01A2 803E0000C8    E     181     ACROSS:                     CMP     YCNT,200D
01A7 7210                182                                 JB      AROUND
01A9 26C606000001  E     183                                 MOV     ES:BAD_MISS,1
01AF C606000001    E     184                                 MOV     ERROR,1
01B4 C6060000C8    E     185                                 MOV     YCNT,200D
01B9 E996FE              186     AROUND:                     JMP     CONTINUE
01BC C606000000    R     187     AGAIN2:                     MOV     BIT_BUCKET,0
01C1 803E120001    R     188                                 CMP     THRUSTER_FIRED,1
01C6 7464                189                                 JE      HERE
01C8 26803E000001  E     190                                 CMP     ES:FIRE_BIRD,1
01CE 755C                191                                 JNE     HERE
```

```
LOC  OBJ                  LINE     SOURCE

01D0 C606120001     R      192                              MOV    THRUSTER_FIRED,1
01D5 C606000001     E      193                              MOV    THRUSTER_FIRE,1
01DA C606110001     R      194                              MOV    BIG_MISS,1
01DF 26C606000001   E      195                              MOV    ES:BAD_MISS,1
01E5 C606000001     R      196                              MOV    BIT_BUCKET,1
01EA E83600                197                              CALL   TIME
01ED 832E010016     R      198                              SUB    ELAPSED_TIME,22D
01F2 2D1600                199                              SUB    AX,22D
01F5 BB0300                200                              MOV    BX,Y_SCALE_FACTOR
01F8 F7E3                  201                              MUL    BX
01FA 05B80B                202                              ADD    AX,3000D
01FD F7360D00       R      203                              DIV    FIFTY
0201 83FA19                204                              CMP    DX,25D
0204 7C01                  205                              JL     GO_ON_ARND
0206 40                    206                              INC    AX
0207 3D0B03                207     GO_ON_ARND:              CMP    AX,779D
020A 7603                  208                              JNA    GO_ON_ARND2
020C B80B03                209                              MOV    AX,779D
020F A30900         R      210     GO_ON_ARND2:             MOV    H_Y_GRAPHIC_POINT,AX
0212 C70607000000   R      211                              MOV    H_X_GRAPHIC_POINT,0
0218 BAD800                212                              MOV    DX,0D8H
021B B01D                  213                              MOV    AL,350
021D E80000         E      214                              CALL   COUT
0220 E902FF                215                              JMP    BY_PASS
                           216
                           217
0223 8306010016     R      218     TIME:                    ADD    ELAPSED_TIME,22D
0228 A10100         R      219                              MOV    AX,ELAPSED_TIME
022B C3                    220                              RET
                           221
022C BAD800                222     HERE:                    MOV    DX,0D8H
022F B01C                  223                              MOV    AL,0340   ;CHANGE RGB TO POINT MODE
0231 E80000         E      224                              CALL   COUT
0234 C7060B000502   R      225                              MOV    HGAE_OFFSET,725D
                           226
023A 803E110001     R      227                              CMP    BIG_MISS,1
023F 7516                  228                              JNE    LEAVE
0241 26C606000000   E      229                              MOV    ES:BAD_MISS,0
0247 C606000000     R      230                              MOV    BIT_BUCKET,0
024C 26C606000000   E      231                              MOV    ES:FIRE_BIRD,0
0252 C606110000     R      232                              MOV    BIG_MISS,0
0257 C606120000     R      233     LEAVE:                   MOV    THRUSTER_FIRED,0
025C C3                    234                              RET
                           235
                           236
----                       237     CODE          ENDS
                           238
                           239                              END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

```
SERIES-III 8086/8087/8088 MACRO ASSEMBLER V1.0 ASSEMBLY OF MODULE MARLYN_MONROE
OBJECT MODULE PLACED IN :F2:RETRO.OBJ
INVOCATION LINE CONTROLS:  DEBUG


LOC  OBJ                  LINE     SOURCE

                             1     NAME    MARLYN_MONROE
                             2     PUBLIC          GRAPH1,COUT,CIN,GRAPH_VREP,GRAPH_HREP,USART_SET_UP_FOR_ADM
                             3     PUBLIC          DELAY_3
                             4
                             5     CGROUP  GROUP   CODE
                             6
                             7     ASSUME  CS:CGROUP
                             8
----                         9     CODE    SEGMENT PUBLIC  'CODE'
                            10
  00D6                      11     TIMER_CNTRL     EQU     0D6H
  00D4                      12     TIMER_CNTRL2    EQU     0D4H
  0086                      13     CNTR2_MODE      EQU     086H
  0004                      14     LOW_ADM         EQU     04H
  0000                      15     HIGH_ADM        EQU     00H
  00DA                      16     USART_CNTRL     EQU     0DAH
  0040                      17     USART_RESET     EQU     40H
  004E                      18     USART_MODE      EQU     4EH
  0037                      19     USART_CMMND     EQU     37H
  00DE                      20     USART_STATUS    EQU     0DEH
                            21
  0075                      22     DELAY_VAL       EQU     75H
                            23
                            24
0000 1D                     25     PICPOINTS       DB      1DH,21H,7CH,2FH,54H,23H,6CH,2FH,54H,440.1440,0650,1270
0001 21
0002 7C
0003 2F
0004 54
0005 23
0006 6C
0007 2F
0008 54
0009 24
000A 64
000B 35
000C 57
000D 36                     26     PICPOINTS1      DB      0660,1570,0660,1120,0660,1570,0660,1360,440,1440,0670,1230,43
                                   0,1540,0750,1260,410
000E 6F
000F 36
0010 4A
0011 36
0012 6F
0013 36
0014 5E
0015 24
0016 64
0017 37
0018 53
```

```
LOC  OBJ                  LINE     SOURCE

0019 23
001A 6C
001B 3D
001C 56
001D 21
001E 7C                     27     PICPOINTS2       DB       1740,0750,1260,410,1740,570,1240,0350,410,1740,0660,1250,0660
                                   ,1570,0660
001F 3D
0020 56
0021 21
0022 7C
0023 2F
0024 54
0025 1D
0026 21
0027 7C
0028 36
0029 55
002A 36
002B 6F
002C 36
002D 55                     28     PICPOINTS3       DB       1250,340,640,1640,660,1230,620,1700,660,1230,600,1740,660,123
                                   0,570,1400,660,1230,550,1440,660,1230,530
002E 1C
002F 34
0030 74
0031 36
0032 53
0033 32
0034 78
0035 36
0036 53
0037 30
0038 7C
0039 36
003A 53
003B 2F
003C 60
003D 36
003E 53
003F 2D
0040 64
0041 36
0042 53
0043 28
0044 68                     29     PICPOINTS4       DB       1500,660,1230,510,1540,660,1230,470,1600,660,1230,450,1640,66
                                   0,1230,430,1700,660,1230,640,1640,660
0045 36
0046 53
0047 29
0048 6C
0049 36
004A 53
004B 27
004C 70
```

```
LOC  OBJ                LINE     SOURCE

004D 36
004E 53
004F 25
0050 74
0051 36
0052 53
0053 23
0054 78
0055 36
0056 53
0057 34
0058 74
0059 36
005A 57                  30      PICPOINTS5      DB      1270,620,1700,660
005B 32
005C 78
005D 36
005E 57                  31      PICPOINTS6      DB      1270,600,1740,660,1270,570,1400,660,1270,550,1440,660,1270,53
                                 0,1500,660,1270,510
005F 30
0060 7C
0061 36
0062 57
0063 2F
0064 60
0065 36
0066 57
0067 2D
0068 64
0069 36
006A 57
006B 2B
006C 68
006D 36
006E 57
006F 29
0070 6C                  32      PICPOINTS7      DB      1540,660,1270,470,1600,660,1270
0071 36
0072 57
0073 27
0074 70
0075 36
0076 57
0077 25                  33      PICPOINTS8      DB      450,1640,660,1270,430,1700,660,1270,350,570,1640,560,1140,370
                                 ,1240
0078 74
0079 36
007A 57
007B 23
007C 78
007D 36
007E 57
007F 1D
0080 2F
0081 74
```

```
LOC  OBJ                  LINE     SOURCE

0082 2E
0083 4C
0084 1F
0085 54
0086 1D                     34     PICPOINTS9      DB      350,560,1660,560,1140,370,1110,350,550,1700,560,1140,370,1150
0087 2E
0088 76
0089 2E
008A 4C
008B 1F
008C 49
008D 1D
008E 2D
008F 78
0090 2E
0091 4C
0092 1F
0093 4D
0094 1D                     35     PICPOINTS10     DB      350,540,1720,560,1140,370,1050,350,520,1760,560,1140,370,1230
                                   ,350,520,1400,560,1140
0095 2C
0096 7A
0097 2E
0098 4C
0099 1F
009A 45
009B 1D
009C 2A
009D 7E
009E 2E
009F 4C
00A0 1F
00A1 53
00A2 1D
00A3 2A
00A4 60
00A5 2E
00A6 4C
00A7 1F                     36     PICPOINTS11     DB      370,1050,350,510,1420,560,1140,370,1030,350,610,1770,560,1220
                                   ,630,1440,560,1220
00A8 45
00A9 1D
00AA 29
00AB 62
00AC 2E
00AD 4C
00AE 1F
00AF 43
00B0 1D
00B1 31
00B2 7F
00B3 2E
00B4 52
00B5 33
00B6 64
```

```
LOC  OBJ                  LINE     SOURCE

00B7 2E
00B8 52
00B9 32                     37     PICPOINTS12     DB      620,1660,560,1300,350,620,1660,560,1140,630,1440,560,1220
00BA 76
00BB 2E
00BC 58
00BD 1D
00BE 32
00BF 76
00C0 2E
00C1 4C
00C2 33
00C3 64
00C4 2E
00C5 52
00C6 1D                     38     PICPOINTS13     DB      350,440,1660,570,1240,370,500,1140,1050,1060,1240,510,350,440
                                   ,1660,710,1310
00C7 24
00C8 76
00C9 2F
00CA 54
00CB 1F
00CC 28
00CD 4C
00CE 45
00CF 46
00D0 54
00D1 29
00D2 1D
00D3 24
00D4 76
00D5 39
00D6 59
00D7 1F                     39     PICPOINTS14     DB      370,500,1220,1110,1070,1100,1240,510
00D8 28
00D9 52
00DA 49
00DB 47
00DC 48
00DD 54
00DE 29
00DF 1C                     40     PICPOINTS15     DB      340,410,1710,570,1240,410,1660,570,1240,410,1710,580,1030,410
                                   ,1660,600,1030
00E0 21
00E1 79
00E2 2F
00E3 54
00E4 21
00E5 76
00E6 2F
00E7 54
00E8 21
00E9 79
00EA 30
00EB 43
```

```
LOC  OBJ                  LINE     SOURCE

00EC 21
00ED 76
00EE 30
00EF 43
00F0 21                     41     PICPOINTS16     DB      410,1710,600,1220,410,1660,600,1220,410,1710,610,1010,410,166
                                   0,610,1010
00F1 79
00F2 30
00F3 52
00F4 21
00F5 76
00F6 30
00F7 52
00F8 21
00F9 79
00FA 31
00FB 41
00FC 21
00FD 76
00FE 31
00FF 41
0100 21                     42     PICPOINTS17     DB      410,1710,610,1200,410,1660,610,1200,410,1710,610,1370,410,166
                                   0,610,1370
0101 79
0102 31
0103 50
0104 21
0105 76
0106 31
0107 50
0108 21
0109 79
010A 31
010B 5F
010C 21
010D 76
010E 31
010F 5F
0110 21                     43     PICPOINTS19     DB      410,1710,620,1160,410,1660,620,1160,410,1710,620,1350,410,166
                                   0,620,1350
0111 79
0112 32
0113 4E
0114 21
0115 76
0116 32
0117 4E
0118 21
0119 79
011A 32
011B 50
011C 21
011D 76
011E 32
011F 50
```

```
LOC  OBJ                  LINE    SOURCE

0120 21                     44    PICPOINTS19    DB       410,1710,630,1140,410,1660,630,1140,410,1710,630,1330,410,166
                                  0,630,1330
0121 79
0122 33
0123 4C
0124 21
0125 76
0126 33
0127 4C
0128 21
0129 79
012A 33
012B 5B
012C 21
012D 76
012E 33
012F 5B
0130 21                     45    PICPOINTS20    DB       410,1710,640,1120,410,1660,640,1120,410,1710,640,1310,410,166
                                  0,640,1310
0131 79
0132 34
0133 4A
0134 21
0135 76
0136 34
0137 4A
0138 21
0139 79
013A 34
013B 59
013C 21
013D 76
013E 34
013F 59
0140 21                     46    PICPOINTS21    DB       410,1710,650,1100,410,1660,650,1100,410,1710,650,1270,410,166
                                  0,650,1270
0141 79
0142 35
0143 48
0144 21
0145 76
0146 35
0147 48
0148 21
0149 79
014A 35
014B 57
014C 21
014D 76
014E 35
014F 57
0150 21                     47    PICPOINTS22    DB       410,1710,660,1060,410,1660,660,1060,410,1710,660,1250,410,166
                                  0,660,1250
0151 79
0152 36
```

```
LOC  OBJ                  LINE     SOURCE

0153 46
0154 21
0155 76
0156 36
0157 46
0158 21
0159 79
015A 36
015B 55
015C 21
015D 76
015E 36
015F 55
0160 21                     48     PICPOINTS23     DB       410,1710,670,1040,410,1660,670,1040,410,1710,670,1230,410,166
                                   0,670,1230
0161 79
0162 37
0163 44
0164 21
0165 76
0166 37
0167 44
0168 21
0169 79
016A 37
016B 53
016C 21
016D 76
016E 37
016F 53
0170 21                     49     PICPOINTS24     DB       410,1710,700,1020,410,1660,700,1020,410,1710,700,1210,410,166
                                   0,700,1210
0171 79
0172 38
0173 42
0174 21
0175 76
0176 38
0177 42
0178 21
0179 79
017A 38
017B 51
017C 21
017D 76
017E 38
017F 51
0180 21                     50     PICPOINTS25     DB       410,1710,710,1000,410,1660,710,1000,410,1710,710,1170,410,166
                                   0,710,1170
0181 79
0182 39
0183 40
0184 21
0185 76
0186 39
```

```
LOC  OBJ                  LINE     SOURCE

0187 40
0188 21
0189 79
018A 39
018B 4F
018C 21
018D 76
018E 39
018F 4F
0190 21                     51     PICPOINTS26     DB     410,1710,710,1360,410,1660,710,1360,410,1710,720,1150,410,166
                                   0,720,1150
0191 79
0192 39
0193 5E
0194 21
0195 76
0196 39
0197 5E
0198 21
0199 79
019A 3A
019B 4D
019C 21
019D 76
019E 3A
019F 4D
01A0 21                     52     PICPOINTS27     DB     410,1710,720,1340,410,1660,720,1340,410,1710,730,1130,410,166
                                   0,730,1130
01A1 79
01A2 3A
01A3 5C
01A4 21
01A5 76
01A6 3A
01A7 5C
01A8 21
01A9 79
01AA 3B
01AB 4B
01AC 21
01AD 76
01AE 3B
01AF 4B
01B0 21                     53     PICPOINTS28     DB     410,1710,730,1320,410,1660,730,1320,410,1710,740,1110,410,166
                                   0,740,1110
01B1 79
01B2 3B
01B3 5A
01B4 21
01B5 76
01B6 3B
01B7 5A
01B8 21
01B9 79
01BA 3C
```

```
LOC  OBJ                  LINE    SOURCE

01BB 49
01BC 21
01BD 76
01BE 3C
01BF 49
01C0 21                     54    PICPOINTS29     DB      410,1710,740,1300,410,1660,740,1300,410,1710,750,1070,410,166
                                  0,750,1070
01C1 79
01C2 3C
01C3 58
01C4 21
01C5 76
01C6 3C
01C7 58
01C8 21
01C9 79
01CA 3D
01CB 47
01CC 21
01CD 76
01CE 3D
01CF 47
01D0 21                     55    PICPOINTS30     DB      410,1710,750,1260,410,1660,750,1260
01D1 79
01D2 3D
01D3 56
01D4 21
01D5 76
01D6 3D
01D7 56
                            56
01D8 10                     57    PICPOINTS31     DB      350,410,1740,420,1140,420,1710,420,1140,440,1650,450,1850,660
                                  ,1570,450,1310
01D9 21
01DA 7C
01DB 22
01DC 4C
01DD 22
01DE 79
01DF 22
01E0 4C
01E1 24
01E2 75
01E3 25
01E4 45
01E5 36
01E6 6F
01E7 25
01E8 59
01E9 36                     58    PICPOINTS32     DB      660,1570,460,1140,440,1650,470,1010,420,1710,550,1230,410,174
                                  0,550,1230
01EA 6F
01EB 26
01EC 4C
01ED 24
```

```
LOC  OBJ                LINE     SOURCE

01EE 75
01EF 27
01F0 41
01F1 22
01F2 79
01F3 20
01F4 53
01F5 21
01F6 70
01F7 20
01F8 53
01F9 21                 59     PICPOINTS33     DB      410,1740,420,1140,350,410,1740,460,1030,660,1570,460,1030
01FA 70
01FB 22
01FC 4C
01FD 10
01FE 21
01FF 70
0200 26
0201 43
0202 36
0203 6F
0204 26
0205 43
0206 10                 60     PICPOINTS34     DB      340,640,1640,460,1010,620,1700,460,1010,600,1740,460,1010,570
                               ,1400,460,1010
0207 34
0208 74
0209 26
020A 41
020B 32
020C 78
020D 26
020E 41
020F 30
0210 70
0211 26
0212 41
0213 2F
0214 60
0215 26
0216 41
0217 30                 61     PICPOINTS35     DB      550,1440,460,1010,530,1500,460,1010,510,1540,460,1010,470,160
                               0,460,1010
0218 64
0219 26
021A 41
021B 2B
021C 66
021D 26
021E 41
021F 29
0220 6C
0221 26
0222 41
```

```
LOC  OBJ                 LINE    SOURCE

0223 27
0224 70
0225 26
0226 41
0227 25                   62     PICPOINTS36      DB      450,1640,460,1010,430,1700,460,1010,640,1640,460,1050,620,170
                                 0,460,1050
0228 74
0229 26
022A 41
022B 23
022C 78
022D 26
022E 41
022F 34
0230 74
0231 26
0232 45
0233 32
0234 78
0235 26
0236 45
0237 30                   63     PICPOINTS37      DB      600,1740,460,1050,570,1400,460,1050,550,1440,460,1050,530,150
                                 0,460,1050
0238 7C
0239 26
023A 45
023B 2F
023C 60
023D 26
023E 45
023F 2D
0240 64
0241 26
0242 45
0243 2B
0244 68
0245 26
0246 45
0247 29                   64     PICPOINTS38      DB      510,1540,460,1050,470,1600,460,1050,450,1640,460,1050,430,170
                                 0,460,1050
0248 6C
0249 26
024A 45
024B 27
024C 70
024D 26
024E 45
024F 25
0250 74
0251 26
0252 45
0253 23
0254 78
0255 26
0256 45
```

```
LOC  OBJ                 LINE     SOURCE

0257 21                    65     PICPOINTS39    DB     410,1710,420,1140,410,1660,420,1140,410,1710,420,1330,410,166
                                  0,420,1330
0258 79
0259 22
025A 40
025B 21
025C 76
025D 22
025E 40
025F 21
0260 79
0261 22
0262 5B
0263 21
0264 76
0265 22
0266 5B
0267 21                    66     PICPOINTS40    DB     410,1710,430,1120,410,1660,430,1120,410,1710,430,1310,410,166
                                  0,430,1310
0268 79
0269 23
026A 4A
026B 21
026C 76
026D 23
026E 4A
026F 21
0270 79
0271 23
0272 59
0273 21
0274 76
0275 23
0276 59
0277 21                    67     PICPOINTS41    DB     410,1710,440,1100,410,1660,440,1100,410,1710,440,1270,410,166
                                  0,440,1270
0278 79
0279 24
027A 48
027B 21
027C 76
027D 24
027E 48
027F 21
0280 79
0281 24
0282 57
0283 21
0284 76
0285 24
0286 57
0287 21                    68     PICPOINTS42    DB     410,1710,450,1060,410,1660,450,1060,410,1710,450,1250,410,166
                                  0,450,1250
0288 79
0289 25
```

```
LOC  OBJ                  LINE     SOURCE

028A 46
028B 21
028C 76
028D 25
028E 46
028F 21
0290 79
0291 25
0292 55
0293 21
0294 76
0295 25
0296 55
0297 21                     69     PICPOINTS43     DB      410,1710,460,1020,410,1660,460,1020,410,1710,460,1230,410,166
                                   0,460,1230
0298 79
0299 26
029A 42
029B 21
029C 76
029D 26
029E 42
029F 21
02A0 79
02A1 26
02A2 53
02A3 21
02A4 76
02A5 26
02A6 53
02A7 21                     70     PICPOINTS44     DB      410,1710,470,1020,410,1660,470,1020,410,1710,470,1210,410,166
                                   0,470,1210
02A8 79
02A9 27
02AA 42
02AB 21
02AC 76
02AD 27
02AE 42
02AF 21
02B0 79
02B1 27
02B2 51
02B3 21
02B4 76
02B5 27
02B6 51
02B7 21                     71     PICPOINTS45     DB      410,1710,500,1000,410,1660,500,1000,410,1710,500,1170,410,166
                                   0,500,1170
02B8 79
02B9 28
02BA 40
02BB 21
02BC 76
02BD 28
```

```
LOC  OBJ                  LINE     SOURCE

02BE 40
02BF 21
02C0 79
02C1 28
02C2 4F
02C3 21
02C4 76
02C5 28
02C6 4F
02C7 21                     72     PICPOINTS46     DB       410,1710,500,1360,410,1660,500,1360,410,1710,510,1150,410,166
                                   0,510,1150
02C8 79
02C9 28
02CA 5E
02CB 21
02CC 76
02CD 28
02CE 5E
02CF 21
02D0 79
02D1 29
02D2 4D
02D3 21
02D4 76
02D5 29
02D6 4D
02D7 21                     73     PICPOINTS47     DB       410,1710,510,1340,410,1660,510,1340,410,1710,520,1130,410,166
                                   0,520,1130
02D8 79
02D9 29
02DA 5C
02DB 21
02DC 76
02DD 29
02DE 5C
02DF 21
02E0 79
02E1 2A
02E2 4B
02E3 21
02E4 76
02E5 2A
02E6 4B
02E7 21                     74     PICPOINTS48     DB       410,1710,520,1320,410,1660,520,1320,410,1710,530,1110,410,166
                                   0,530,1110
02E8 79
02E9 2A
02EA 5A
02EB 21
02EC 76
02ED 2A
02EE 5A
02EF 21
02F0 79
02F1 2B
```

```
LOC  OBJ                LINE     SOURCE

02F2 49
02F3 21
02F4 76
02F5 2B
02F6 49
02F7 21                  75     PICPOINTS49    DB      410,1710,530,1300,410,1660,530,1300,410,1710,540,1070,410,166
                                0,540,1070
02F8 79
02F9 2B
02FA 58
02FB 21
02FC 76
02FD 2B
02FE 58
02FF 21
0300 79
0301 2C
0302 47
0303 21
0304 76
0305 2C
0306 47
0307 21                  76     PICPOINTS50    DB      410,1710,540,1260,410,1660,540,1260,410,1710,550,1050,410,166
                                0,550,1050
0308 79
0309 2C
030A 56
030B 21
030C 76
030D 2C
030E 56
030F 21
0310 79
0311 2D
0312 45
0313 21
0314 76
0315 2D
0316 45
0317 21                  77     PICPOINTS51    DB      410,1710,550,1230,410,1660,550,1230
0318 79
0319 2D
031A 53
031B 21
031C 76
031D 2D
031E 53
031F 10                  78     PICPOINTS52    DB      350,440,1660,420,1140,370,500,1250,1200,510,350,440,1660,500,
                                1240
0320 24
0321 76
0322 22
0323 4C
0324 1F
0325 28
```

```
LOC  OBJ                  LINE     SOURCE

0326 55
0327 50
0328 29
0329 1D
032A 24
032B 76
032C 28
032D 54
032E 1F                     79     PICPOINTS53     DB      370,500,1040,1170,1270,1160,510,150,300
032F 28
0330 44
0331 4F
0332 57
0333 4E
0334 29
0335 0D
0336 18
                            80
                            81
                            82
0337                        83     LAST    LABEL   WORD
                            84
                            85     ;***************************************************************************
                            86     ;                            START VERTICAL REP  DATA
                            87     ;***************************************************************************
                            88
                            89     ;START VERTICAL ENVELOPE
0337 1D                     90     VREP1           DB      350,410,1740,620,1350,420,1720,620,1350,440,1660,710,
                                                        1170
0338 21
0339 7C
033A 32
033B 5D
033C 22
033D 7A
033E 32
033F 5D
0340 24
0341 76
0342 39
0343 4F
0344 36                     91     VREP2           DB      660,1570,720,1040,660,1570,720,1260,440,1660,730,
                                                        1130,420,1720,760,1030
0345 6F
0346 3A
0347 44
0348 36
0349 6F
034A 3A
034B 56
034C 24
034D 76
034E 3B
034F 4B
0350 22
```

```
LOC  OBJ                LINE     SOURCE

0351 7A
0352 3E
0353 4C
0354 21                   92     VREP3          DB      410,1740,760,1030,410,1740,620,1350,350,410,1740,720,
                                                    1150,660,1570,720,1150
0355 7C
0356 3E
0357 4C
0358 21
0359 7C
035A 32
035B 5D
035C 1D
035D 21
035E 7C
035F 3A
0360 4D
0361 36
0362 6F
0363 3A
0364 4D
                          93     ;END VERT ENVELOPE
                          94     ;START WORDS "UP,DOWN" ON RIGHT SIDE OF SCREEN
0365 1C                   95     VREP4          DB      350,440,1660,750,1210,370,500,1250,1200,510,350,440,
                                                    1660,650,1340,370,500,1040,1170,1270,1160,510
0366 24
0367 76
0368 3D
0369 51
036A 1F
036B 28
036C 55
036D 50
036E 29
036F 1D
0370 24
0371 76
0372 35
0373 5C
0374 1F
0375 28
0376 44
0377 4F
0378 57
0379 4E
037A 29
                          96     ;END
                          97     ;START VERTICAL REP MIL TICS
037B 1C                   98     VREP5          DB      340,410,1710,620,1350,410,1660,620,1350,410,1710,630,
                                                    1140,410,1660
037C 21
037D 79
037E 32
037F 5D
0380 21
```

```
LOC  OBJ                  LINE     SOURCE

0381 76
0382 32
0383 5D
0384 21
0385 79
0386 33
0387 4C
0388 21
0389 76
038A 33                     99     VREP55         DB      630,1140,410,1710,630,1330,410,1660,630,1330
038B 4C
038C 21
038D 79
038E 33
038F 5B
0390 21
0391 76
0392 33
0393 5B
0394 21                    100     VREP6          DB      410,1710,640,1120,410,1660,640,1120,410,1710,640,1310,
                                                      410,1660,640,1310
0395 79
0396 34
0397 4A
0398 21
0399 76
039A 34
039B 4A
039C 21
039D 79
039E 34
039F 59
03A0 21
03A1 76
03A2 34
03A3 59
03A4 21                    101     VREP7          DB      410,1710,650,1160,410,1660,650,1160,410,1710,650,1270,
                                                      410,1660,650,1270
03A5 79
03A6 35
03A7 48
03A8 21
03A9 76
03AA 35
03AB 48
03AC 21
03AD 79
03AE 35
03AF 57
03B0 21
03B1 76
03B2 35
03B3 57
03B4 21                    102     VREP8          DB      410,1710,660,1060,410,1660,660,1060,410,1710,660,1250,
                                                      410,1660,660,1250
```

```
LOC  OBJ                  LINE     SOURCE

03B5 79
03B6 36
03B7 46
03B8 21
03B9 76
03BA 36
03BB 46
03BC 21
03BD 79
03BE 36
03BF 55
03C0 21
03C1 76
03C2 36
03C3 55
03C4 21                   103     VREP9         DB      410,1710,670,1040,410,1660,670,1040,410,1710,670,1230,
                                                            410,1660,670,1230
03C5 79
03C6 37
03C7 44
03C8 21
03C9 76
03CA 37
03CB 44
03CC 21
03CD 79
03CE 37
03CF 53
03D0 21
03D1 76
03D2 37
03D3 53
03D4 21                   104     VREP10        DB      410,1710,700,1020,410,1660,700,1020,410,1710,700,1210,
                                                            410,1660,700,1210
03D5 79
03D6 38
03D7 42
03D8 21
03D9 76
03DA 38
03DB 42
03DC 21
03DD 79
03DE 38
03DF 51
03E0 21
03E1 76
03E2 38
03E3 51
03E4 21                   105     VREP11        DB      410,1710,710,1000,410,1660,710,1000,410,1710,710,1170,
                                                            410,1660,710,1170
03E5 79
03E6 39
03E7 40
03E8 21
```

```
LOC  OBJ                 LINE     SOURCE

03E9 76
03EA 39
03EB 40
03EC 21
03ED 79
03EE 39
03EF 4F
03F0 21
03F1 76
03F2 39
03F3 4F
03F4 21                   106     VREP12          DB      410,1710,710,1360,410,1660,710,1360,410,1710,720,1150,
                                                       410,1660,720,1150
03F5 79
03F6 39
03F7 5E
03F8 21
03F9 76
03FA 39
03FB 5E
03FC 21
03FD 79
03FE 3A
03FF 4D
0400 21
0401 76
0402 3A
0403 4D
0404 21                   107     VREP13          DB      410,1710,720,1340,410,1660,720,1340,410,1710,730,1130,
                                                       410,1660,730,1130
0405 79
0406 3A
0407 5C
0408 21
0409 76
040A 3A
040B 5C
040C 21
040D 79
040E 3B
040F 4B
0410 21
0411 76
0412 3B
0413 4B
0414 21                   108     VREP14          DB      410,1710,730,1320,410,1660,730,1320,410,1710,740,1110,
                                                       410,1660,740,1110
0415 79
0416 3B
0417 5A
0418 21
0419 76
041A 3B
041B 5A
041C 21
```

```
LOC  OBJ                LINE    SOURCE

041D 79
041E 3C
041F 49
0420 21
0421 76
0422 3C
0423 49
0424 21             109     VREP15         DB      410,1710,740,1300,410,1660,740,1300,410,1710,750,1070
0425 79
0426 3C
0427 5
0428 21
0429 76
042A 3C
042B 58
042C 21
042D 79
042E 3D
042F 47
0430 21             110     VREP05         DB      410,1660,750,1070,410,1710,750,1260,410,1660,750,1260
0431 76
0432 3D
0433 47
0434 21
0435 79
0436 3D
0437 56
0438 21
0439 76
043A 3D
043B 56
                    111     ;END
                    112     ;START TIME TICS
043C 10             113     VREP16         DB      340,640,1640,720,1130,620,1700,720,1130,600,1740,720,
                                                1130,570,1400,720,1130
043D 34
043E 74
043F 3A
0440 4B
0441 1
0442 8
0443 3A
0444 4B
0445 30
0446 7C
0447 3A
0448 4B
0449 2F
044A 6
044B 3A
044C 4B
044D 3D             114     VREP17         DB      550,1440,720,1130,530,1500,720,1130,510,1540,720,1130,
                                                470,1600,720,1130
044E 64
044F 3A
```

```
LOC  OBJ                  LINE     SOURCE

0450 4F
0451 2B
0452 6B
0453 3A
0454 4B
0455 29
0456 6C
0457 3A
0458 4B
0459 2C
045A 7A
045B 3A
045C 4B
045D 25                   115      VREF18         DB       450,1640,720,1130,430,1700,720,1130,640,1640,720,1170,
                                                       620,1700,720,1170
045E 74
045F 3A
0460 4B
0461 2C
0462 78
0463 3A
0464 4B
0465 24
0466 74
0467 3A
0468 4F
0469 32
046A 7E
046B 7A
046C 4F
046D 30                   116      VREF19         DE       600,1740,720,1170,570,1400,720,1170,550,1440,720,1170,
                                                       530,1500,720,1170
046E 7C
046F 3A
0470 4F
0471 2F
0472 60
0473 7A
0474 4F
0475 2D
0476 64
0477 39
0478 4F
0479 2B
047A 68
047B 3A
047C 4F
047D 29                   117      VREF20         DB       510,1540,720,1170,470,1600,720,1170,450,1640,720,1170,
                                                       430,1700,720,1170
047E 6C
047F 3A
0480 4F
0481 27
0482 70
0483 3A
```

```
LOC  OBJ                 LINE    SOURCE

0484 4F
0485 2F
0486 74
0487 3A
0488 4F
0489 25
048A 73
048B 3A
048C 4F
                         118     .END
                         119     ;START "TIME-SEC ARROW"
048D 1D                  120     VREP21          DB      350,570,1640,560,1140,370,1240,350,560,1660,560,1140,
                                                           370,1110,350,550,1700,560,1140,370,1150
048E 2F
048F 74
0490 2E
0491 4C
0492 1F
0493 54
0494 1D
0495 2E
0496 76
0497 2E
0498 4C
0499 1F
049A 49
049B 1D
049C 2D
049D 78
049E 2E
049F 4C
04A0 1F
04A1 4D
04A2 1D                  121     VREP22          DB      350,540,1720,560,1140,370,1050,350,520,1760,560,1140
04A3 2C
04A4 7A
04A5 2E
04A6 4C
04A7 1F
04A8 45
04A9 1D
04AA 2A
04AB 74
04AC 2E
04AD 4C
04AE 1F                  122     VREP02          DB      370,1230,350,520,1400,560,1140,370,1050,350,510,1420
04AF 53
04B0 1D
04B1 2A
04B2 6A
04B3 2E
04B4 4C
04B5 1F
04B6 45
04B7 1D
```

```
LOC  OBJ                  LINE     SOURCE

04B8 29
04B9 62
04BA 2E                    123     VREP23         DB      560,1140,370,1030
04BB 40
04BC 1F
04BD 41
04BE 1D                    124     VREF24         DB      350,610,1770,560,1220,630,1440,560,1220,620,1660,560
04BF 71
04C0 7F
04C1 2E
04C2 52
04C3 33
04C4 64
04C5 2E
04C6 52
04C7 32
04C8 76
04C9 2E
04CA 58                    125     VREP04         DB      1300,350,620,1660,560,1140,630,1440,560,1220
04CB 1D
04CC 32
04CD 76
04CE 2E
04CF 40
04D0 33
04D1 64
04D2 2E
04D3 52
                           126     ;END
                           127     ;START LEFT ENVELOPE
04D4 1D                    128     VREP25         DB      350,410,1740,560,1110,410,1740,420,1000,660,1570,420,
                                                     1000
04D5 21
04D6 7C
04D7 2E
04D8 43
04D9 21
04DA 7C
04DB 22
04DC 40
04DD 36
04DE 6F
04DF 22
04E0 40
                           129     ;END
                           130     ;START "MISSILE ABOVE GROUND (FEET)"
04E1 1D                    131     VREF26         DB      350,400,1760,420,1100,370,1150,1110,1230,1230
                                                     1110,1140,1050,400,1010,1020,1170,1260,1050
04E2 20
04E3 7E
04E4 22
04E5 48
04E6 1F
04E7 40
04E8 49
```

```
LOC  OBJ                LINE     SOURCE

04E9 53
04EA 53
04EB 43
04EC 4C
04ED 45
04EE 20
04EF 41
04F0 42
04F1 4F
04F2 56
04F3 45
04F4 20                 132     VREP06          DB      400,1070,1220,1170,1250,1160,1040,400,500,1060,1050,
                                                          1050,1240,510,350
04F5 47
04F6 52
04F7 4F
04F8 55
04F9 4E
04FA 44
04FB 20
04FC 28
04FD 46
04FE 45
04FF 45
0500 54
0501 29
0502 10
                        133     .END
                        134     .START "GROUND LEVEL"
0503 1D                 135     VREP86          DB      350,630,1440,410,1100,370 1070,350,620,1460,410,1100,
                                                   370,1220,350,610,1500,410,1100,370,1170
0504 33
0505 64
0506 21
0507 48
0508 1F
0509 47
050A 1D
050B 32
050C 66
050D 21
050E 48
050F 1F
0510 52
0511 1D
0512 31
0513 68
0514 21
0515 48
0516 1F
0517 4F
0518 1D                 136     VREP27          DB      350,600,1520,410,1100,370,1250,350,570,1540,410,1100,
                                                   370,1160,350,560,1560,410,1100,370,1040
0519 30
051A 6A
```

```
LOC  OBJ                 LINE     SOURCE

051B 21
051C 48
051D 1F
051E 55
051F 1D
0520 2F
0521 6C
0522 21
0523 48
0524 1F
0525 4E
0526 1D
0527 2E
0528 6E
0529 21
052A 48
052B 1F
052C 44
052D 1D                  137      VREP28          DB      350,540,1620,410,1100,370,1140,350,530,1640,410,1100,
                                                      370,1050,350,520,1660,410,1100,370,1260
052E 2C
052F 72
0530 21
0531 48
0532 1F
0533 4C
0534 1D
0535 2B
0536 74
0537 21
0538 48
0539 1F
053A 45
053B 1D
053C 2A
053D 76
053E 21
053F 48
0540 1F
0541 56
0542 1D                  138      VREP29          DB      350,510,1700,410,1100,370,1050,350,500,1720,410,1100,
                                                      370,1140
0543 29
0544 78
0545 21
0546 48
0547 1F
0548 45
0549 1D
054A 28
054B 7A
054C 21
054D 48
054E 1F
054F 4C
```

```
LOC  OBJ                  LINE     SOURCE

                           139     ;END
                           140     ;START "FEET TICS"
0550 1C                    141     VREP30          DB      34O,41O,171O,42O,100O,41O,166O,42O,100O,41O,171O,43O,
                                                        114O,41O,166O,43O,114O
0551 21
0552 79
0553 22
0554 40
0555 21
0556 76
0557 22
0558 40
0559 21
055A 79
055B 23
055C 4C
055D 21
055E 76
055F 23
0560 4C
0561 21                    142     VREP31          DB      41O,171O,44O,127O,41O,166O,44O,127O,41O,171O,46O,103O,
                                                        41O,166O,46O,103O
0562 79
0563 24
0564 57
0565 21
0566 76
0567 24
0568 57
0569 21
056A 79
056B 26
056C 43
056D 21
056E 76
056F 26
0570 43
0571 21                    143     VREP32          DB      41O,171O,47O,117O,41O,166O,47O,117O,41O,171O,50O,132O,
                                                        41O,166O,50O,132O
0572 79
0573 27
0574 4F
0575 21
0576 76
0577 27
0578 4F
0579 21
057A 79
057B 28
057C 5A
057D 21
057E 76
057F 28
0580 5A
0581 21                    144     VREP33          DB      41O,171O,52O,106O,41O,166O,52O,106O,41O,171O,53O,122O,
LOC  OBJ                  LINE     SOURCE

                                                  410,1660,530,1220
0582 79
0583 2A
0584 46
0585 21
0586 76
0587 2A
0588 46
0589 21
058A 79
058B 2B
058C 52
058D 21
058E 76
058F 2B
0590 52
0591 21                   145      VREP34          DB      410,1710,540,1350,410,1660,540,1350,410,1710,560,1110,
                                                  410,1660,560,1110
0592 79
0593 2C
0594 50
0595 21
0596 76
0597 2C
0598 50
0599 21
059A 79
059B 2E
059C 49
059D 21
059E 76
059F 2E
05A0 49
                          146
                          147      ;*********************************************************************************
                          148      ;                      THESE NEXT DATA ARE FOR HORIZONTAL REPRIESE
                          149      ;*********************************************************************************
                          150
                          151      ;END
                          152      ;START "0 AND 3"
05A1 1D                   153      VREP35          DB      350,420,1410,420,1120,370,600,350,420,1410,460,1030,
                                                  370,630
05A2 22
05A3 61
05A4 22
05A5 4A
05A6 1F
05A7 30
05A8 1D
05A9 22
05AA 61
05AB 26
05AC 43
05AD 1F
05AE 33
```

```
LOC  OBJ                 LINE     SOURCE

                          154     ;END
                          155     ;START "GUNNER AIMING ERROR (MILS)"
05AF 1D                   156     VREP36          DB      350,400,1760,620,1350,370,1070,1250,1160,1160,1050,1220
                                                      400,1010,1110,1150,1110,1160,1070,400,1050
05B0 20
05B1 7F
05B2 32
05B3 5D
05B4 1F
0555 47
05B6 55
05B7 4E
05B8 4E
05B9 45
05BA 52
05BB 20
05BC 41
05BD 49
05BE 4D
05BF 49
05C0 4E
05C1 47
05C2 20
05C3 45
05C4 52                   157     VREP37          DB      1220,1220,1170,1220,400,500,1150,1110,1140,1230,510,350
05C5 52
05C6 4F
05C7 52
05C8 20
05C9 28
05CA 4D
05CB 49
05CC 4C
05CD 53
05CE 29
05CF 1D
                          158     ;END
                          159
05D0                      160     LAST_OF_VREP    LABEL   WORD
                          161
                          162     ;START HORIZONTAL GAE ENVELOPE
                          163
05D0 1D                   164     HREP    DB      350,410,1740,570,1240,430,1540,570,1240,440,1440,0650,1270
05D1 21
05D2 7C
05D3 2F
05D4 54
05D5 23
05D6 6C
05D7 2F
05D8 54
05D9 24
05DA 64
05DB 35
05DC 57
```

```
LOC  OBJ                LINE     SOURCE

05DD 36                  165     HREP1    DB      0660,1570,0660,1120,0660,1570,0660,1360,440,1440,0670,1230,430,1540,0
                                 750,1260,410
05DE 6F
05DF 36
05E0 4A
05E1 36
05E2 6F
05E3 36
05E4 5E
05E5 24
05E6 64
05E7 37
05E8 53
05E9 23
05EA 6C
05EB 3D
05EC 56
05ED 21
05EE 7C                  166     HREP2    DB      1740,0750,1260,410,1740,570,1240,0350,410,1740,0660,1250,0660,1570,06
                                 60
05EF 3D
05F0 56
05F1 21
05F2 7C
05F3 2F
05F4 54
05F5 1D
05F6 21
05F7 7C
05F8 36
05F9 55
05FA 36
05FB 6F
05FC 36
                         167     ;END
                         168     ;START TIME TICS
05FD 55                  169     HREP3    DB      1250,340,540,1640,660,1230,620,1700,660,1230,600,1740,660,1230,570,14
                                 00,660,1230,550,1440,660,1230,530
05FE 1C
05FF 34
0600 74
0601 36
0602 53
0603 32
0604 78
0605 36
0606 53
0607 30
0608 7C
0609 36
060A 53
060B 2F
060C 60
060D 36
060E 53
```

```
LOC  OBJ                 LINE     SOURCE

060F 2D
0610 64
0611 36
0612 53
0613 2B
0614 68                   170     HREP4   DB      1500,660,1230,510,1540,660,1230,470,1600,660,1230,450,1640,660,1230,4
                                  30,1700,660,1230,640,1640,660
0615 36
0616 53
0617 29
0618 6C
0619 36
061A 53
061B 27
061C 70
061D 36
061E 53
061F 25
0620 74
0621 36
0622 53
0623 23
0624 78
0625 36
0626 53
0627 34
0628 74
0629 36
062A 57                   171     HREP5   DB      1270,620,1700,660
062B 32
062C 78
062D 36
062E 57                   172     HREP6   DB      1270,600,1740,660,1270,570,1400,660,1270,550,1440,660,1270,530,1500,6
                                  60,1270,510
062F 30
0630 7C
0631 36
0632 57
0633 2F
0634 60
0635 36
0636 57
0637 2D
0638 64
0639 36
063A 57
063B 2B
063C 68
063D 36
063E 57
063F 29
0640 6C                   173     HREP7   DB      1540,660,1270,470,1600,660,1270
0641 36
0642 57
0643 27
```

```
LOC  OBJ                 LINE     SOURCE

0644 70
0645 36
0646 57
                          174     ;END SOMEWHERE NEAR HERE
                          175     ;START ARROW
0647 25                   176     HREP8   DB      450,1640,660,1270,430,1700,660,1270,350,570,1640,560,1140,370,1240
0648 74
0649 36
064A 57
064B 25
064C 76
064D 36
064E 57
064F 1D
0650 2F
0651 74
0652 2E
0653 4C
0654 1F
0655 54
0656 1D                   177     HREP9   DB      350,560,1660,560,1140,370,1110,350,550,1700,560,1140,370,1150
0657 2E
0658 76
0659 2E
065A 4C
065B 1F
065C 49
065D 1D
065E 2D
065F 78
0660 2E
0661 4C
0662 1F
0663 4D
0664 1D                   178     HREP10  DB      350,540,1720,560,1140,370,1050,350,520,1760,560,1140,370,1230,350,520
                                  ,1400,560,1140
0665 2C
0666 7A
0667 2E
0668 4C
0669 1F
066A 45
066B 1D
066C 2A
066D 7E
066E 2E
066F 4C
0670 1F
0671 53
0672 1D
0673 2A
0674 60
0675 2E
0676 4C
0677 1F                   179     HREP11  DB      370,1050,350,510,1420,560,1140,370,1030,350,610,1770,560,1220,630,144
```

```
LOC  OBJ                LINE     SOURCE

                                 0,560,1220
0678 45
0679 1D
067A 29
067B 62
067C 2E
067D 4C
067E 1F
067F 43
0680 1D
0681 31
0682 7F
0683 2E
0684 52
0685 33
0686 64
0687 2E
0688 52
0689 32                   180    HREP12  DB      620,1660,560,1300,350,620,1660,560,1140,630,1440,560,1220
068A 76
068B 2E
068C 58
068D 1D
068E 32
068F 76
0690 2E
0691 4C
0692 33
0693 64
0694 2E
0695 52
0696 1D                   181    HREP13  DB      350,440,1660,570,1240,370,500,1140,1050,1060,1240,510,350,440,1660,71
                                 0,1310
0697 24
0698 76
0699 2F
069A 54
069B 1F
069C 28
069D 4C
069E 45
069F 46
06A0 54
06A1 29
06A2 1D
06A3 24
06A4 76
06A5 39
06A6 59
06A7 1F                   182    HREP14  DB      370,500,1220,1110,1070,1100,1240,510
06A8 28
06A9 52
06AA 49
06AB 47
06AC 48
```

```
LOC  OBJ                  LINE     SOURCE

06AD 54
06AE 29
                           183     ;END
                           184     ;START MIL TICS
06AF 1C                    185     HREP15  DB      340,410,1710,570,1240,410,1660,570,1240,410,1710,600,1030,410,1660,60
                                   0,1030
06B0 21
06B1 79
06B2 2F
06B3 54
06B4 21
06B5 76
06B6 2F
06B7 54
06B8 21
06B9 79
06BA 30
06BB 43
06BC 21
06BD 76
06BE 30
06BF 43
06C0 21                    186     HREP16  DB      410,1710,600,1220,410,1660,600,1220,410,1710,610,1010,410,1660,610,10
                                   10
06C1 79
06C2 30
06C3 52
06C4 21
06C5 76
06C6 30
06C7 52
06C8 21
06C9 79
06CA 31
06CB 41
06CC 21
06CD 76
06CE 31
06CF 41
06D0 21                    187     HREP17  DB      410,1710,610,1200,410,1660,610,1200,410,1710,610,1370,410,1660,610,13
                                   70
06D1 79
06D2 31
06D3 50
06D4 21
06D5 76
06D6 31
06D7 50
06D8 21
06D9 79
06DA 31
06DB 5F
06DC 21
06DD 76
06DE 31
```

```
LOC  OBJ                  LINE    SOURCE

06DF 5F
06E0 21                    188    HREP18  DB      410,1710,620,1160,410,1660,620,1160,410,1710,620,1350,410,1660,620,13
                                  50
06E1 79
06E2 32
06E3 4E
06E4 21
06E5 76
06E6 32
06E7 4E
06E8 21
06E9 79
06EA 32
06EB 5D
06EC 21
06ED 76
06EE 32
06EF 5D
06F0 21                    189    HREP19  DB      410,1710,630,1140,410,1660,630,1140,410,1710,630,1330,410,1660,630,13
                                  30
06F1 79
06F2 33
06F3 4C
06F4 21
06F5 76
06F6 33
06F7 4C
06F8 21
06F9 79
06FA 33
06FB 5B
06FC 21
06FD 76
06FE 33
06FF 5B
0700 21                    190    HREP20  DB      410,1710,640,1120,410,1660,640,1120,410,1710,640,1310,410,1660,640,13
                                  10
0701 79
0702 34
0703 4A
0704 21
0705 76
0706 34
0707 4A
0708 21
0709 79
070A 34
070B 59
070C 21
070D 76
070E 34
070F 59
0710 21                    191    HREP21  DB      410,1710,650,1100,410,1660,650,1100,410,1710,650,1270,410,1660,650,12
                                  70
0711 79
```

```
LOC  OBJ                  LINE    SOURCE

0712 35
0713 48
0714 21
0715 76
0716 35
0717 48
0718 21
0719 79
071A 35
071B 57
071C 21
071D 76
071E 35
071F 57
0720 21                    192    HREP22  DB      410,1710,660,1060,410,1660,660,1060,410,1710,660,1250,410,1660,660,12
                                  50
0721 79
0722 36
0723 46
0724 21
0725 76
0726 36
0727 46
0728 21
0729 79
072A 36
072B 55
072C 21
072D 76
072E 36
072F 55
0730 21                    193    HREP23  DB      410,1710,670,1040,410,1660,670,1040,410,1710,670,1230,410,1660,670,12
                                  30
0731 79
0732 37
0733 44
0734 21
0735 76
0736 37
0737 44
0738 21
0739 79
073A 37
073B 53
073C 21
073D 76
073E 37
073F 57
0740 21                    194    HREP24  DB      410,1710,700,1020,410,1660,700,1020,410,1710,700,1210,410,1660,700,12
                                  10
0741 79
0742 38
0743 42
0744 21
0745 76
```

```
LOC  OBJ                  LINE     SOURCE

0746 38
0747 42
0748 21
0749 79
074A 38
074B 51
074C 21
074D 76
074E 38
074F 51
0750 21                    195     HREP25  DB      410,1710,710,1000,410,1660,710,1000,410,1710,710,1170,410,1660,710,11
                                   70
0751 79
0752 39
0753 40
0754 21
0755 76
0756 39
0757 40
0758 21
0759 79
075A 39
075B 4F
075C 21
075D 76
075E 39
075F 4F
0760 21                    196     HREP26  DB      410,1710,710,1360,410,1660,710,1360,410,1710,720,1150,410,1660,720,11
                                   50
0761 79
0762 39
0763 5E
0764 21
0765 76
0766 39
0767 5E
0768 21
0769 79
076A 3A
076B 40
076C 21
076D 76
076E 3A
076F 40
0770 21                    197     HREP27  DB      410,1710,720,1340,410,1660,720,1340,410,1710,730,1130,410,1660,730,11
                                   30
0771 79
0772 3A
0773 5C
0774 21
0775 76
0776 3A
0777 5C
0778 21
0779 79
```

```
LOC  OBJ                    LINE     SOURCE

077A 3B
077B 4B
077C 21
077D 76
077E 3B
077F 4B
0780 21                      198     HREP28  DB       410,1710,730,1320,410,1660,730,1320,410,1710,740,1110,410,1660,740,11
                                     10
0781 79
0782 3B
0783 5A
0784 21
0785 76
0786 3B
0787 5A
0788 21
0789 79
078A 3C
078B 49
078C 21
078D 76
078E 3C
078F 49
0790 21                      199     HREP29  DB       410,1710,740,1300,410,1660,740,1300,410,1710,750,1070,410,1660,750,10
                                     70
0791 79
0792 3C
0793 58
0794 21
0795 76
0796 3C
0797 58
0798 21
0799 79
079A 3D
079B 47
079C 21
079D 76
079E 3D
079F 47
07A0 21                      200     HREP30  DB       410,1710,750,1260,410,1660,750,1260
07A1 79
07A2 3D
07A3 56
07A4 21
07A5 76
07A6 3D
07A7 56
                             201     ;END MIL TICS
                             202
                             203     ;START "GUNNER AIMING ERROR (MILS)"
07A8 1D                      204     HREP31          DB       350,400,1760,600,1220,370,1070,1250,1160,1160,1050,1220,
                                                          400,1010,1110,1150,1110,1160,1070,400,1050
07A9 20
07AA 7E
```

```
LOC  OBJ                  LINE     SOURCE

07AB 30
07AC 52
07AD 1F
07AE 47
07AF 55
07B0 4E
07B1 4E
07B2 45
07B3 52
07B4 20
07B5 41
07B6 49
07B7 4D
07B8 49
07B9 4E
07BA 47
07BB 20
07BC 45
07BD 52                   205      HREP32         DB      1220,1220,1170,1220,400,500,1150,1110,1140,1230,510,350
07BE 52
07BF 4F
07C0 52
07C1 20
07C2 28
07C3 4D
07C4 49
07C5 4C
07C6 53
07C7 29
07C8 1D
                          206      ;END
                          207
                          208      ;START "FEET TICS"
07C9 1C                   209      HREP33         DB      340,410,1710,420,1000,410,1660,420,1000,410,1710,430,
                                                       1140,410,1660,430,1140
07CA 21
07CB 79
07CC 22
07CD 40
07CE 21
07CF 76
07D0 22
07D1 40
07D2 21
07D3 79
07D4 23
07D5 4C
07D6 21
07D7 76
07D8 23
07D9 4C
07DA 21                   210      HREP34         DB      410,1710,440,1270,410,1660,440,1270,410,1710,460,1030,
                                                       410,1660,460,1030
07DB 79
07DC 24
```

```
LOC  OBJ                  LINE    SOURCE

07DD 57
07DE 21
07DF 76
07E0 24
07E1 57
07E2 21
07E3 79
07E4 26
07E5 43
07E6 21
07E7 76
07E8 26
07E9 43
07EA 21                    211     HREP35       DB      410,1710,470,1170,410,1660,470,1170,410,1710,500,1320,
                                                     410,1660,500,1320
07EB 79
07EC 27
07ED 4F
07EE 21
07EF 76
07F0 27
07F1 4F
07F2 21
07F3 79
07F4 28
07F5 5A
07F6 21
07F7 76
07F8 28
07F9 5A
07FA 21                    212     HREP36       DB      410,1710,520,1060,410,1660,520,1060,410,1710,530,1220,
                                                     410,1660,530,1220
07FB 79
07FC 2A
07FD 46
07FE 21
07FF 76
0800 2A
0801 46
0802 21
0803 79
0804 2B
0805 52
0806 21
0807 76
0808 2B
0809 52
080A 21                    213     HREP37       DB      410,1710,540,1350,410,1660,540,1350
080B 79
080C 2C
080D 50
080E 21
080F 76
0810 2C
0811 5D
```

```
LOC  OBJ                LINE     SOURCE

                         214     ;END
                         215
                         216     ;START LEFT ENVELOPE
                         217
0812 1D                  218     HRE38          DB      350,410,1740,540,1350,410,1740,420,1000,350,410,1740,
                                                     470,1170,660,1570,470,1170
0813 21
0814 7C
0815 2C
0816 5D
0817 21
0818 7C
0819 22
081A 40
081B 1D
081C 21
081D 7C
081E 27
081F 4F
0820 36
0821 6F
0822 27
0823 4F
                         219     ;END
                         220     ;START "TARGET LINE"
0824 1D                  221     HREP38         DB      350,640,1520,460,1270,370,1240,350,630,1540,460,1270,
                                                     370,1010,350,620,1600,460,1270,370,1220
0825 34
0826 6A
0827 26
0828 57
0829 1F
082A 54
082B 1C
082C 33
082D 6C
082E 26
082F 57
0830 1F
0831 41
0832 1D
0833 32
0834 70
0835 26
0836 57
0837 1F
0838 52
0839 1D                  222     HREP39         DB      350,610,1600,460,1270,370,1070,350,600,1620,460,1270,
                                                     370,1050,350,570,1640,460,1270,370,1240
083A 31
083B 70
083C 26
083D 57
093E 1F
083F 47
```

```
LOC  OBJ                 LINE    SOURCE

0840 1D
0841 30
0842 72
0843 26
0844 57
0845 1F
0846 45
0847 1D
0848 2F
0849 74
084A 26
084B 57
084C 1F
084D 54
084E 1D                   223    HREP40          DB       350,630,1540,470,1320,370,1140,350,620,1600,470,1320,370
                                                        ,1110,350,610,1600,470,1320,370,1160
084F 33
0850 6C
0851 27
0852 5A
0853 1F
0854 4C
0855 1D
0856 32
0857 70
0858 27
0859 5A
085A 1F
085B 49
085C 1D
085D 31
085E 70
085F 27
0860 5A
0861 1F
0862 4E
0863 1D                   224    HREP41          DB       350,600,1620,470,1320,370,1050
0864 30
0865 72
0866 27
0867 5A
0868 1F
0869 45
                          225    ;END
                          226    ;START "(FEET)"
                          227
086A 1D                   228    HREP42          DB       350,400,1760,460,1150,370,500,1060,1050,1050,1240,510
086B 20
086C 7E
086D 26
086E 4D
086F 1F
0870 28
0871 46
0872 45
```

```
LOC  OBJ               LINE     SOURCE

0873 45
0874 54
0875 29
                        229     ;END
                        230     ;START "(LEFT).(RIGHT)"
0876 1D                 231     HREP43          DB      350,440,1660,420,1000,370,500,1140,1050,1060,1240,510,
                                                     350,440,1660,520,1060,370,500,1220,1110

0877 24
0878 76
0879 22
087A 40
087B 1F
087C 28
087D 4C
087E 45
087F 46
0880 54
0881 29
0882 1D
0883 24
0884 76
0885 2A
0886 46
0887 1F
0888 28
0889 52
088A 49
088B 47                 232     HREP44          DB      1070,1100,1240,510,350
088C 48
088D 54
088E 29
088F 1D
                        233
0890                    234     LAST_OF_HREP    LABEL   WORD
                        235
                        236
0890 E88600             237     GRAPH1:                 CALL CLEAR_SCREEN
                        238
0893 BE0000      R      239     GRAPH:                  MOV     SI,OFFSET PICPOINTS
0896 B93703             240                             MOV     CX,(OFFSET LAST-OFFSET PICPOINTS)
0899 BAD800             241                             MOV     DX,0D8H
089C 2E8A04             242     AGAIN:                  MOV     AL,CS:[SI]
089F E86400             243                             CALL    COUT
08A2 46                 244                             INC     SI
08A3 E88A00             245                             CALL    DELAY_2
08A6 E2F4               246                             LOOP    AGAIN
08A8 C3                 247                             RET
                        248
08A9 E86D00             249     GRAPH_VREP:             CALL    CLEAR_SCREEN
08AC BE3703      R      250                             MOV     SI,OFFSET VREP1
08AF B99902             251                             MOV     CX,(OFFSET LAST_OF_VREP-OFFSET VREP1)
08B2 BAD800             252                             MOV     DX,0D8H
08B5 2E8A04             253     AGN_N_AGN               MOV     AL,CS:[SI]
08B8 E84800             254                             CALL    COUT
08BE 46                 255                             INC     SI
```

```
LOC  OBJ             LINE    SOURCE

08BC E87100                                          CALL    DELAY_2
08BF E2F4                                            LOOP    AGN_N_AGN
08C1 C3                                              RET

08C2 E85400                  GRAPH_HREP:             CALL    CLEAR_SCREEN
08C5 BED005        R                                 MOV     SI,OFFSET HREP
08C8 B9C002                                          MOV     CX,(OFFSET LAST_OF_HREP-OFFSET HREP)
08CB BAD800                                          MOV     DX,0D8H
08CE 2E8A04                  AGN2_N_AGN              MOV     AL,CS:[SI]
08D1 E83200                                          CALL    COUT
08D4 46                                              INC     SI
08D5 E85800                                          CALL    DELAY_2
08D8 E2F4                                            LOOP    AGN2_N_AGN
08DA C3                                              RET



08DB BAD600                  USART_SET_UP_FOR_ADM:   MOV     DX,TIMER_CNTRL
08DE B0B6                                            MOV     AL,CNTR2_MODE
08E0 EE                                              OUT     DX,AL
08E1 BAD400                                          MOV     DX,TIMER_CNTRL2
08E4 B004                                            MOV     AL,LOW_ADM
08E6 E84100                                          CALL    DELAY
08E9 EE                                              OUT     DX,AL
08EA B000                                            MOV     AL,HIGH_ADM
08EC E83B00                                          CALL    DELAY
08EF EE                                              OUT     DX,AL
08F0 BADA00                                          MOV     DX,USART_CNTRL
08F3 B040                                            MOV     AL,USART_RESET
08F5 E83200                                          CALL    DELAY
08F8 EE                                              OUT     DX,AL
08F9 B04E                                            MOV     AL,USART_MODE
08FB E82C00                                          CALL    DELAY
08FE EE                                              OUT     DX,AL
08FF B037                                            MOV     AL,USART_CMMND
0901 E82600                                          CALL    DELAY
0904 EE                                              OUT     DX,AL
0905 C3                                              RET

0906 50                      COUT:                   PUSH    AX
0907 E4DE                    OVER1:                  IN      AL,USART_STATUS
0909 2401                                            AND     AL,1
090B 74FA                                            JZ      OVER1
090D 58                                              POP     AX
090E EE                                              OUT     DX,AL
090F C3                                              RET

0910 E4DE                    CIN:                    IN      AL,USART_STATUS
0912 2402                                            AND     AL,2
0914 74FA                                            JZ      CIN
0916 E4DC                                            IN      AL,0DCH
0918 C3                                              RET

0919 BAD800                  CLEAR_SCREEN            MOV     DX,0D8H  ;THIS PROCEDURE CLEARS THE
091C B01D                                            MOV     AL,350   ;ADM-3 SCREEN
```

```
LOC  OBJ                LINE   SOURCE

091E E8E5FF              311                          CALL   COUT
0921 B019                312                          MOV    AL,31O
0923 E8E0FF              313                          CALL   COUT
0926 E81100              314                          CALL   DELAY_3 ;THE CLEAR SCREEN MODE TAKES
                         315                                 ;160MS TO COMPLETE
0929 C3                  316                          RET
                         317
092A B90300              318   DELAY:                 MOV    CX,3
092D E2FE                319   TAG:                   LOOP   TAG
092F C3                  320                          RET
                         321
0930 B87500              322   DELAY_2:               MOV    AX,DELAY_VAL
0933 48                  323   AGAIN_AND_AGAIN:       DEC    AX
0934 3D0000              324                          CMP    AX,0
0937 75FA                325                          JNE    AGAIN_AND_AGAIN
0939 C3                  326                          RET
                         327
093A B850C3              328   DELAY_3:               MOV    AX,50000D
093D 48                  329   OVER_AND_OVER:         DEC    AX
093E 3D0000              330                          CMP    AX,0
0941 75FA                331                          JNE    OVER_AND_OVER
0943 B850C3              332                          MOV    AX,50000D
0946 48                  333   ONE_MORE_TIME:         DEC    AX
0947 3D0000              334                          CMP    AX,0
094A 75FA                335                          JNE    ONE_MORE_TIME
094C C3                  336                          RET
                         337
----                     338   CODE                   ENDS
                         339
                         340                          END

ASSEMBLY COMPLETE, NO ERRORS FOUND
```

APPENDIX D

COMPUTER GENERATED SOUND SYSTEMS PROGRAMS

```
ASM48 SOUNDZ SRC

ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0           PAGE    1

 LOC  OBJ         SEQ        SOURCE STATEMENT

                   1 ;                          1 SEPTEMBER 1981
                   2
                   3
                   4 ;       THIS PROGRAM WRITTEN IN INTEL 8748 ASSEMBLY LANGUAGE IS THE SOURCE
                   5 ; FOR SOUNDS PRODUCED DURING A SIMULATED DRAGON FIRING AND RESIDES IN EPROM
                   6 ; OF THE SOUND SYSTEM MICROCOMPUTER (SSM).  THESE SOUNDS ARE:
                   7
                   8 ;       1.  GYRO WIND UP
                   9 ;       2.  LAUNCH EXPLOSION
                  10 ;       3.  THRUSTER FIRINGS
                  11 ;       4.  IMPACT EXPLOSIONS
                  12 ;             A. TARGET HIT
                  13 ;             B. TARGET MISSED
                  14
                  15
                  16 ;       THE DFS PROCESSOR SENDS A FOUR BIT WORD WHICH IS DECODED BY THE
                  17 ; SSM INTO ONE OF THE ABOVE SOUNDS  THE SSM IN TURN PASSES DATA BYTES TO
                  18 ; A GENERAL INSTRUMENTS AY-3-8910 PROGRAMMABLE SOUND GENERATOR (PSG).  THE
                  19 ; PSG INTERPRETS 14 DATA BYTES STORED IN THE LOWER 14 LOCATIONS OF AN ON
                  20 ; CHIP 16 BYTE REGISTER ARRAY AS A SOUND, THEREBY GENERATING AN ANALOG
                  21 ; SIGNAL.
                  22
                  23 ;**********************************************************************
                  24
                  25 ;       INITIALIZATION ROUNTINE
                  26
0000              27         ORG     0
0000 0409         28         JMP     ANIT
                  29
0003              30         ORG     3
0003 04D1         31         JMP     DECODE
                  32
0007              33         ORG     7
0007 16C6         34         JTF     TIMER
                  35
0009              36         ORG     09H
0009 446D         37 ANIT:   JMP     CAKAP
000B 65           38 INIT:   STOP    TCNT
000C 27           39         CLR     A
000D 62           40         MOV     T,A             ;INITIALIZE TIMER AND ENABLE INTERRUPT
000E 25           41         EN      TCNTI
                  42
000F 05           43         EN      I               ;ENABLE INTERRUPT
0010 75           44         ENT0    CLK             ;ENABLE CLOCK ON T0
0011 23FF         45         MOV     A,#0FFH
0013 39           46         OUTL    P1,A            ;INITIALIZE PORTS 1 & 2
0014 3A           47         OUTL    P2,A
                  48
0015 230B         49         MOV     A,#0BH          ;INITIALIZE PORTS 4 & 5
0017 3C           50         MOVD    P4,A
0018 27           51         CLR     A
0019 3D           52         MOVD    P5,A
```

```
ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0             PAGE    2

 LOC  OBJ         SEQ        SOURCE STATEMENT

                   53
001A 996F          54 RESPG:  ANL     P1,#06FH        ;RESET PSG'S
001C 8990          55         ORL     P1,#090H
                   56
                   57
001E 9ABF          58 FLIP:   ANL     P2,#0BFH        ;SET FLIP/FLOPS
0020 8A40          59         ORL     P2,#040H
                   60
                   61
                   62 ;***********************************************************************
                   63
                   64 ;       LOOP ROUTINE I.E. WAIT FOR INTERRUPT
                   65
0022 85            66         CLR     F0
0023 27            67         CLR     A
0024 B93E          68         MOV     R1,#03EH
0026 B82F          69         MOV     R0,#02FH
0028 A0            70         MOV     @R0,A
0029 C8            71         DEC     R0
002A A0            72         MOV     @R0,A
002B C8            73         DEC     R0
002C A0            74         MOV     @R0,A
002D A1            75         MOV     @R1,A
002E 19            76         INC     R1
002F A1            77         MOV     @R1,A
0030 AB            78         MOV     R3,A
0031 AC            79         MOV     R4,A
0032 AD            80         MOV     R5,A
0033 D5            81         SEL     RB1
0034 A8            82         MOV     R0,A
0035 A9            83         MOV     R1,A
0036 AA            84         MOV     R2,A
0037 AB            85         MOV     R3,A
0038 AC            86         MOV     R4,A
0039 AD            87         MOV     R5,A
003A AE            88         MOV     R6,A
003B AF            89         MOV     R7,A
003C C5            90         SEL     RB0
                   91
003D B82F          92 DRANG:  MOV     R0,#02FH
003F F0            93         MOV     A,@R0
0040 C63D          94         JZ      DRANG
0042 27            95         CLR     A
0043 A0            96         MOV     @R0,A
0044 5428          97         CALL    DRAGON
                   98
0046 B82F          99 INFIN:  MOV     R0,#02FH
0048 F0           100         MOV     A,@R0
0049 C646         101         JZ      INFIN
                  102
004B 1494         103         CALL    CHECK
                  104
004D B820         105         MOV     R0,#020H
004F F0           106         MOV     A,@R0
0050 AC           107         MOV     R4,A            ;R4 GETS TIME
```

D-2

```
LOC  OBJ          SEQ        SOURCE STATEMENT

0051 54B8          108          CALL    ROPOP
0053 C676          109          JZ      SINFIN
                   110
0055 AF            111          MOV     R7,A              ;R7 GETS COUNT
0056 B921          112          MOV     R1,#021H
0058 F1            113          MOV     A,@R1
0059 AA            114          MOV     R2,A
005A 14B4          115          CALL    DELAY
005C AA            116          MOV     R2,A
005D 14B4          117          CALL    DELAY
005F B820          118          MOV     R0,#020H
0061 B921          119          MOV     R1,#021H
                   120
0063 F1            121 UNFIN:   MOV     A,@R1
0064 20            122          XCH     A,@R0
0065 EF72          123          DJNZ    R7,ANFIN
0067 B646          124          JF0     INFIN
0069 D5            125          SEL     RB1
006A 27            126          CLR     A
006B A6            127          MOV     R0,A
006C AB            128          MOV     R3,A
006D C5            129          SEL     RB0
006E 85            130          CLR     F0
006F 95            131          CPL     F0
0070 0446          132          JMP     INFIN
                   133
0072 18            134 ANFIN:   INC     R0
0073 19            135          INC     R1
0074 0463          136          JMP     UNFIN
                   137
0076 B82E          138 SINFIN:  MOV     P0,#02EH          ;LOCATION OF RB1R3 PPEVIOUS
0078 F0            139          MOV     A,@R0
0079 C67E          140          JZ      SAFON
007B AA            141          MOV     P2,A
007C 14BF          142          CALL    DALAY
007E B820          143 SAFON:   MOV     R0,#020H          ;LOCATION OF RB1R0 PREVIOUS
0080 F0            144          MOV     A,@R0
0081 C687          145          JZ      SAVOB
0083 BAFF          146          MOV     R2,#0FFH
0085 14BF          147          CALL    DALAY
0087 D5            148 SAVOB:   SEL     RB1
0088 FB            149          MOV     A,P3
0089 C5            150          SEL     RB0
008A B82E          151          MOV     R0,#02EH
008C A0            152          MOV     @R0,A
008D D5            153          SEL     PB1
008E F8            154          MOV     A,R0
008F C5            155          SEL     PB0
0090 C8            156          DEC     R0
0091 A0            157          MOV     @P0,A
0092 0446          158          JMP     INFIN
                   159
0094 07            160 CHECK:   DEC     A
0095 96AA          161          JNZ     CHAK
0097 B83E          162          MOV     R0,#03EH
```

```
LOC  OBJ         SEQ        SOURCE STATEMENT

0099 F0           163          MOV     A,@R0
009A 96A1         164          JNZ     DEXO
009C 18           165          INC     R0
009D F0           166          MOV     A,@R0
009E 96AB         167          JNZ     HEXO
00A0 93           168 CHAK:    RETR
                  169
00A1 18           170 DEXO     INC     R0
00A2 F0           171          MOV     A,@R0
00A3 C6A7         172          JZ      DEXAA
00A5 446D         173          JMP     CAKAP
00A7 14AF         174 DEXAA:   CALL    GETTA
00A9 4477         175          JMP     DUDEX
                  176
00AB 14AF         177 HEXO:    CALL    GETTA
00AD 443A         178          JMP     HITEX
                  179
00AF B820         180 GETTA:   MOV     R0,#020H
00B1 F0           181          MOV     A,@R0
00B2 AC           182          MOV     R4,A
00B3 93           183          RETR
                  184
                  185
                  186 ;**********************************************************************
                  187
                  188 ;                DELAY    SUBROUTINES
                  189
                  190
00B4 B908         191 DELAY:   MOV     R1,#8           ;DELAY = R2 X .01 SEC
00B6 B8FF         192 LOOP1    MOV     R0,#0FFH
00B8 E8B8         193 LOOP2:   DJNZ    R0,LOOP2
00BA E9B6         194          DJNZ    R1,LOOP1
00BC EAB4         195          DJNZ    R2,DELAY
00BE 93           196          RETR
                  197
00BF B8C8         198 DALAY:   MOV     R0,#0C8H        ;DALAY = R2 X .001 SEC
00C1 E8C1         199 DLY      DJNZ    R0,DLY
00C3 EABF         200          DJNZ    R2,DALAY
00C5 93           201          RETR
                  202
                  203
                  204 ;**********************************************************************
                  205
                  206 ;        TIMER INTERRUPT SERVICE ROUTINE
                  207
                  208 ;        THE TIMER INCREMENTS EVERY 80USEC   (T)
                  209 ;        R4 INCREMENTS EVERY 20MSEC
                  210 ;        R5 INCREMENTS EVERY 5.24 SEC
                  211
                  212
00C6 D5           213 TIMER:   SEL     RB1
00C7 AD           214          MOV     R5,A
00C8 1C           215          INC     R4
                  216
00C9 1E           217          INC     R6
```

```
                    SOURCE STATEMENT

218            MOV     A,R6
219            JNZ     DELO
220            INC     R7
221 DELO       MOV     A,R5
222            SEL     RB0
223            RETR
224
225
226 ;*******************************************************************
227
228 ;          EXTERNAL INTERRUPT SERVICE ROUTINE   DECODE LOWER 3 BITS OF 4 BIT
229 ;          WORD FROM DFS PROCESSOR   FLIP/FLOP'S  LATCH WORD ONTO PORT 1
230
231
232 DECODE     SEL     RB1
233            MOV     R5,A
234            IN      A,P1
235            ANL     A,#7
236
237            JZ      DRAGO          ;        000     INTI BANG
238            DEC     A
239            MOV     R1,#03EH
240            JZ      DUDE           ;        001     GND DUD
241            DEC     A
242            JZ      DUDEX1         ;        010     GND EXPLO
243            DEC     A
244            JZ      DUDEX1         ;        011     MISS EXPLO
245            DEC     A
246            JZ      HITE1          ;        100     HIT EXPLO
247            DEC     A
248            JZ      ROPO1          ;        101     ROCK POP
249
250            JMP     PUFF
251
252 DUDEX1     MOVX    A,@R0          ; STOP TIMER
253            JMP     DUDEXO
254 HITE1      MOVX    A,@R0          ; STOP TIMER
255            JMP     HITE
256 ROPO1      JMP     ROPO
257
258 DRAGO      MOV     R1,#02FH       ;        PSG#1
259            INC     @R1
260            JMP     PUPPET
261 DUDE       INC     @R1
262            INC     R1
263            INC     @R1
264            MOV     R1,#02FH
265            INC     @R1
266            JMP     PUFF           ;        PSG#2
267 DUDEXO     INC     @R1
268            JMP     ROPO           ;        PSG#2
269 HITE       INC     R1
270            INC     @R1                     PSG#2
271 ROPO       MOV     R1,#02FH
272            INC     @R1
```

```
LOC  OBJ         SEQ        SOURCE STATEMENT

                 273
0108 F1          274        MOV    A,@R1
0109 AA          275        MOV    R2,A             ;R2 GETS COUNT
010A B920        276        MOV    R1,#020H         ;R1 GETS ADDRESS
010C FC          277        MOV    A,R4             ;ACC GETS TIME
                 278
010D EA12        279 PUSS:  DJNZ   R2,POSS
010F A1          280        MOV    @R1,A
0110 2415        281        JMP    PASS
                 282
0112 19          283 POSS:  INC    R1
0113 240D        284        JMP    PUSS
                 285
0115 BC00        286 PASS:  MOV    R4,#0
0117 FD          287 PUPP:  MOV    A,R5
0118 C5          288        SEL    RB0
0119 93          289        RETR
                 290
011A 861E        291 PUPPET JNI    POKER
011C 2417        292        JMP    PUPP
011E 2308        293 POKER  MOV    A,#8
0120 D7          294        MOV    PSW,A
0121 15          295        DIS    I
0122 BA64        296        MOV    R2,#100D
0124 14B4        297        CALL   DELAY
0126 040E        298        JMP    INIT
                 299
                 300
                 301 ;***************************************************************************
                 302
                 303 ;      SUBROUTINE TO CHECK TIMER AND GIVE ASSOCIATED DELAY AND AMPLITUDE
                 304 ;      DECAY
                 305
0128 C5          306 TSTR45: SEL   RB0
0129 BB00        307        MOV    R3,#0
012B BE0F        308        MOV    R6,#0FH
012D BF00        309        MOV    R7,#0
012F D5          310        SEL    RB1              ;CUMULATIVE DELAY SO FAR
0130 FB          311        MOV    A,R3
0131 C5          312        SEL    RB0
0132 C637        313        JZ     TARC
0134 AA          314        MOV    R2,A
0135 146F        315        CALL   DALAY
0137 D5          316 TARC:  SEL    RB1
0138 F8          317        MOV    A,R0
0139 C5          318        SEL    RB0
013A C640        319        JZ     TESTR4
013C BAFF        320        MOV    R2,#0FFH
013E 146F        321        CALL   DALAY
0140 244C        322        JMP    TESTR4
                 323 ;                             IF R5 = 1 THEN FLIGHT TIME = 5.243SEC
                 324 ;                             AT LEAST. THEN DELAY = 1.352SEC
                 325
                 326 ;                             IF R5 = 2 THEN FLIGHT TIME = 10.486SEC
                 327                                AT LEAST. THEN DELAY = 2.704SEC
```

```
[illegible]            SEQ        SOURCE STATEMENT

                       328
                       329
0142 97                330 TAR     CLR     C
014  FE                331         MOV     A,R3          PB0R3 KEEPS INCREMENTAL DELAY
0144 D5                332         SEL     RB1
0145 6B                333         ADD     A,R3
0146 AE                334         MOV     R3,A         .RB1R3  HOLDS CUMULATIVE DELAY
0147 E64A              335         INC     TAPB
0149 18                336         INC     R0
014A [illegible]       337 TAKE    SEL     RB0
014B [illegible]       338         RETR
                       339
                       340
[illegible]            341 TESTR4  SEL     RB0
[illegible]            342         MOV     A,R4
[illegible]            343         J67     BIT7
0150 [illegible]       344 JP6     JB6     BIT6
[illegible] B77B       345 JP5     JB5     BIT5
[illegible]            346 JP4     JB4     BIT4
[illegible] 7291       347 JP3     JB3     BIT3
[illegible]            348 JP2     JB2     BIT2
[illegible]            349 JP1     JB1     BIT1
[illegible]            350 JP0     JB0     BIT0
015E [illegible]       351         JMP     TARAP
                       352
[illegible]            353 BIT7    MOV     R2,#67D       .FOR BIT 7 TRAVEL = 2 621SEC
[illegible] 14E4       354         CALL    DELAY         .THEREFORE DELAY =  676SEC
0164 BA06              355         MOV     R2,#6
0166 14EF              356         CALL    DALAY
0168 BB44              357         MOV     R3,#68D
016A 2450              358         JMP     JP6
                       359
016C BA21              360 BIT6    MOV     R2,#33D       .FOR BIT 6 TRAVEL = 1 311SEC
                       361 .                              THEREFORE DELAY =  338SEC
016E 14E4              362         CALL    DELAY
01 0 BA08              363         MOV     R2,#8
01 2 14BF              364         CALL    DALAY
0174 FB                365         MOV     A,R3
0175 0322              366         ADD     A,#340
0177 AB                367         MOV     R3,A
0178 FC                368         MOV     A,R4
0179 [illegible]       369         JMP     JP5
                       370
017B BAA9              371 BIT5    MOV     R2,#169D      .FOR BIT 5 TRAVEL =  655SEC
                       372 .                              THEREPEFORE DELAY =  169SEC
[illegible] 14EF       373         CALL    DALAY
[illegible]            374         MOV     A,R3
0180 0311              375         ADD     A,#170
0182 AB                376         MOV     R3,A
[illegible]            377         MOV     A,R4
0184 7454              378         JMP     JP4
                       379
0186 [illegible]       380 BIT4    MOV     R2,#84D       .TRAVEL =  329SEC   DELAY =  084SEC
[illegible] 14[illegible]          CALL    DALAY
018A FB [illegible]                MOV     A,R3
```

```
LOC  OBJ         SEQ        SOURCE STATEMENT

018B 039D         383          ADD     A,#9D
018D AB           384          MOV     R3,A
018E FC           385          MOV     A,R4
018F 2456         386          JMP     JP3
                  387
0191 BA2A         388 BIT3:    MOV     R2,#42D          ;TRAVEL =  164SEC  DELAY = .042SEC
0193 146F         389          CALL    DALAY
0195 FB           390          MOV     A,R3
0196 0304         391          ADD     A,#4
0198 AB           392          MOV     R3,A
0199 FC           393          MOV     A,R4
019A 2458         394          JMP     JP2
                  395
019C BA15         396 BIT2:    MOV     R2,#21D          ;TRAVEL =  082SEC  DELAY =  021SEC
019E 146F         397          CALL    DALAY
01A0 1B           398          INC     R3
01A1 1B           399          INC     R3
01A2 FC           400          MOV     A,R4
01A3 245A         401          JMP     JP1
                  402
01A5 BA0B         403 BIT1:    MOV     R2,#11D          ;TRAVEL =  041SEC  DELAY = .011SEC
01A7 146F         404          CALL    DALAY
01A9 1B           405          INC     R3
01AA FC           406          MOV     A,R4
01AB 245C         407          JMP     JP0
                  408
01AD BA05         409 BIT0:    MOV     R2,#5D           ;TRAVEL =  020SEC  DELAY =  005SEC
01AF 146F         410          CALL    DALAY
01B1 1B           411          INC     R3
                  412
                  413
                  414 ;        AMPLITUDE DECAY IS DEPENDENT ON THE TOTAL TIME STORED
                  415 ;        IN REGISTERS RB1R6 (20 MSEC PER BIT) AND RB1R7.
                  416
                  417
01B2 D5           418 TARAP:   SEL     RB1
01B3 FF           419          MOV     A,R7
01B4 C5           420          SEL     RB0
01B5 C6C0         421          JZ      TRAP
01B7 07           422          DEC     A
01B8 C6BE         423          JZ      TRIP
01BA BE02         424          MOV     R6,#2            ;TRAVEL=10 486 SEC
01BC 2442         425          JMP     TAR
01BE BE08         426 TRIP:    MOV     R6,#8
01C0 D5           427 TRAP:    SEL     RB1
01C1 FE           428          MOV     A,R6
01C2 C5           429          SEL     RB0
01C3 F2DF         430          JB7     BAT7
01C5 D2E6         431 PB6:     JB6     BAT6
01C7 B2EF         432 PB5:     JB5     BAT5
01C9 92F7         433 PB4:     JB4     BAT4
01CB 72FF         434 PB3:     JB3     BAT3
01CD 5205         435 PB2:     JB2     BAT21
01CF 3207         436 PB1:     JB1     BAT11
01D1 1209         437 PB0:     JB0     BAT01
```

```
LOC  OBJ         LINE        SOURCE STATEMENT

01D3 44[illegible]        438              JMP     TRAMF
                  439
01D5 4407         440 BAT21    JMP     BAT2
01D7 440F         441 BAT11    JMP     BAT1
01D9 4415         442 BAT01    JMP     BAT0
                  443
01DB D5           444 PBY      SEL     RB1
01DC FE           445          MOV     A,R6
01DD C5           446          SEL     RB0
01DE 93           447          RETR
                  448
01DF CE           449 BAT7     DEC     R6
01E0 CE           450          DEC     R6
01E1 CE           451          DEC     R6
01E2 BF2A         452          MOV     R7,#42D
01E4 24[illegible]         453          JMP     PB6
                  454
01E6 CE           455 BAT6     DEC     R6
01E7 2339         456          MOV     A,#57D
01E9 6F           457          ADD     A,R7
01EA AF           458          MOV     R7,A
01EB 34DB         459          CALL    PBY
01ED 24[illegible]         460          JMP     PB5
                  461
01EF 23[illegible]         462 BAT5     MOV     A,#66D
01F1 6F           463          ADD     A,R7
01F2 AF           464          MOV     R7,A
01F3 34DB         465          CALL    PBY
01F5 24[illegible]         466          JMP     PB4
                  467
01F7 23[illegible]         468 BAT4     MOV     A,#33D
01F9 6F           469          ADD     A,R7
01FA AF           470          MOV     R7,A
01FB 34DB         471          CALL    PBY
01FD 24[illegible]         472          JMP     PB3
                  473
01FF 2310         474 BAT3     MOV     A,#16D
0201 6F           475          ADD     A,R7
0202 AF           476          MOV     R7,A
0203 34DB         477          CALL    PBY
0205 24[illegible]         478          JMP     PB2
                  479
0207 2308         480 BAT2     MOV     A,#8
0209 6F           481          ADD     A,R7
020A AF           482          MOV     R7,A
020B 34DB         483          CALL    PBY
020D 24[illegible]         484          JMP     PB1
                  485
020F 1F           486 BAT1     INC     R7
0210 1F           487          INC     R7
0211 1F           488          INC     R7
0212 1F           489          INC     R7
0213 24[illegible]         490          JMP     PB0
                  491
0215 1F           492 BAT0     INC     R7
```

```
LOC  OBJ         SEQ        SOURCE STATEMENT

0216 1F           493         INC     R7
                  494
                  495
0217 FF           496 TRAMP:  MOV     A,R7
                  497
                  498 ;       FOR FURTHER AMPLITUDE CONTROL R7 IS NOW TESTED
                  499
0218 F21E         500         JB7     BEA
021A D225         501 TAP1    JB6     BEB
021C 2442         502         JMP     TAR
                  503
021E CE           504 BEA:    DEC     R6
021F 537F         505         ANL     A,#07FH         ;  - 1280MSEC
0221 0336         506         ADD     A,#54D          ;  +  540MSEC
0223 441A         507         JMP     TAP1
                  508
0225 CE           509 BEB:    DEC     R6
0226 2442         510         JMP     TAR
                  511
                  512
                  513 ;*************************************************************************
                  514
                  515 ;       ROUTINE TO INITIATE INTERNAL TIMER & GIVE
                  516 ;       INITIAL LAUNCH EXPLOSION
                  517
                  518
0228 55           519 DRAGON: STRT    T
                  520
0229 80           521         MOVX    A,@R0           ;START TIMER
                  522
022A 996F         523         ANL     P1,#06FH                 ;RESET PSG'S
022C 8990         524         ORL     P1,#090H
                  525
022E 9ADF         526         ANL     P2,#0DFH                 ;SELECT NO. 1
                  527
                  528 ;       LOAD  PSG #1
                  529
                  530
0230 B820         531         MOV     R0,#020H
0232 BFC0         532         MOV     R7,#0300Q                ;BANG ON CHANNELS A,B & C  OF PSG#1
                  533
0234 5438         534         CALL    LODRAM
                  535
0236 7400         536         CALL    LOADP1
                  537
0238 8A30         538         ORL     P2,#030H                 ;TRI-STATE
                  539
023A 93           540         RETR
                  541
023B 235D         542 LODRAM: MOV     A,#135Q
023D A0           543         MOV     @R0,A           ;R0
023E 18           544         INC     R0
023F 230D         545         MOV     A,#15Q
0241 A0           546         MOV     @R0,A           ;R1
0242 18           547         INC     R0
```

```
LOC  OBJ         SEQ        SOURCE STATEMENT

0243 235D        548          MOV     A,#135Q
0245 A0          549          MOV     @P0,A          ;R2
0246 18          550          INC     P0
0247 230D        551          MOV     A,#15Q
0249 A0          552          MOV     @R0,A          ;R3
024A 18          553          INC     R0
024B 235D        554          MOV     A,#135Q
024D A0          555          MOV     @R0,A          ;R4
024E 18          556          INC     R0
024F 230D        557          MOV     A,#15Q
0251 A0          558          MOV     @R0,A          ;R5
0252 18          559          INC     R0
0253 231F        560          MOV     A,#37Q
0255 A0          561          MOV     @R0,A          ;R6
0256 18          562          INC     R0
0257 FF          563          MOV     A,F7
0258 A0          564          MOV     @P0,A          ;R7
0259 18          565          INC     F0
025A 2310        566          MOV     A,#20Q
025C A0          567          MOV     @P0,A          ;R8
025D 18          568          INC     R0
025E A0          569          MOV     @R0,A          ;P9
025F 18          570          INC     P0
0260 A0          571          MOV     @F0,A          ;R10
0261 18          572          INC     R0
0262 23FF        573          MOV     A,#0FFH
0264 A0          574          MOV     @P0,A          ;R11
0265 18          575          INC     R0
0266 233F        576          MOV     A,#077Q
0268 A0          577          MOV     @R0,A          ;R12
0269 18          578          INC     P0
026A 27          579          CLP     A
026B A0          580          MOV     @R0,A          ;R13
026C 83          581          PETR
                 582
                 583
                 584 ;************************************************************
                 585
                 586 ;        THIS ROUTINE WAITS FOR A SYSTEM RESET TO START A NEW FLIGHT.
                 587
026D 2308        588 LAUNF    MOV     A,#8
026F D7          589          MOV     PSW,A
0270 15          590          DIS     I
0271 8675        591 BRAVE    JNI     COKOP
0273 4471        592          JMP     BRAVE
0275 04AA        593 COKOF    JMP     INIT
                 594
                 595
                 596 ;************************************************************
                 597
                 598          THIS ROUTINE PPODUCES A DOUBLE EXPLOSION WHEN MISSILE
                 599          HAS MISSED THE TAPGET
                 600
0277 9AEF        601 DUDEX    ANL     P2,#0EFH        ;SELECT PSG #2
                 602
```

```
LOC  OBJ          SEQ        SOURCE STATEMENT

0279 3428          603            CALL    TSTR45          ;CALL DELAY AND AMPLITUDE
                   604
027B BD02          605            MOV     R5,#2           ;LOOP COUNTER FOR AMP DECAY
                   606
027D A5            607            CLR     F1
                   608
027E 397F          609 CAKET      ANL     P1,#07FH
0280 8980          610            ORL     P1,#080H        ;RESET PSG #2
                   611
0282 B830          612            MOV     R0,#030H
0284 BFC9          613            MOV     R7,#3110        ;MISS ON CHANNEL B & C ONLY (PSG#2)
                   614
0286 543B          615            CALL    LODRAM
0288 7492          616            CALL    HITAMP
028A 7423          617            CALL    LOADP2
                   618
028C BAFF          619            MOV     R2,#0FFH
028E 146F          620            CALL    DALAY
                   621
0290 7696          622            JF1     JIERT
0292 B5            623            CPL     F1
0293 CD            624            DEC     R5
0294 447E          625            JMP     CAKET
                   626
0296 9A30          627 JIERT      ORL     P2,#030H        ;TRI-STATE
                   628
0298 446D          629            JMP     CAKAP
                   630
                   631
                   632 ;**********************************************************************
                   633
                   634 ;          THIS POUTINE PRODUCES A TRIPLE EXPLOSION WHEN MISSILE
                   635 ;          HITS THE TARGET.
                   636
029A BD03          637 HITEX:     MOV     R5,#3           ;LOOP COUNTER
                   638
029C 3428          639            CALL    TSTR45          ;CALL DELAY AND AMPLITUDE
                   640
029E 9AEF          641 KAKOT      ANL     P2,#0EFH        ;SELECT PSG #2
                   642
02A0 397F          643            ANL     P1,#07FH
02A2 8980          644            ORL     P1,#080H        ;RESET PSG #2
                   645
02A4 B830          646            MOV     R0,#030H
02A6 EFC9          647            MOV     R7,#3110        ;HIT ON CHANNEL B & C ONLY (PSG#2)
                   648
02A8 543B          649            CALL    LODRAM
02AA 7492          650            CALL    HITAMP
02AC 7423          651            CALL    LOADP2
                   652
02AE 8A30          653            ORL     P2,#030H
                   654
02B0 BAFF          655            MOV     R2,#0FFH
02B2 146F          656            CALL    DALAY
                   657
```

```
LOC  OBJ         SEQ        SOURCE STATEMENT

02B4 ED4E        658        DJNZ    R5,KAKOT
                 659
02B6 44ED        660        JMP     CAKAP
                 661
                 662
                 663 ;************************************************************************
                 664
                 665 ;      THIS ROUTINE PRODUCES A SOUND SIMULATING THE IGNITION OF A
                 666        SIDE ROCKET THRUSTER PAIR.
                 667
02B8 [illegible] 668 ROPOP  CALL    TSTF45
                 669
02BA 997F        670        ANL     P1,#07FH
02BC 8980        671        ORL     P1,#080H               ;RESET PSG #2
                 672
02BE 9AEF        673        ANL     P2,#0EFH               ;SELECT PSG #2
                 674
02C0 B830        675        MOV     R0,#030H
02C2 235D        676        MOV     A,#135Q
02C4 A0          677        MOV     @R0,A          ;R0
02C5 18          678        INC     R0
02C6 230D        679        MOV     A,#15Q
02C8 A0          680        MOV     @R0,A          ;R1
02C9 18          681        INC     R0
02CA 235D        682        MOV     A,#135Q
02CC A0          683        MOV     @R0,A          ;R2
02CD 18          684        INC     R0
02CE 230D        685        MOV     A,#15Q
02D0 A0          686        MOV     @R0,A          ;R3
02D1 18          687        INC     R0
02D2 235D        688        MOV     A,#135Q
02D4 A0          689        MOV     @R0,A          ;R4
02D5 18          690        INC     R0
02D6 230D        691        MOV     A,#15Q
02D8 A0          692        MOV     @R0,A          ;R5
02D9 18          693        INC     R0
02DA 231F        694        MOV     A,#37Q
02DC A0          695        MOV     @R0,A          ;R6
02DD 18          696        INC     R0
02DE 23F6        697        MOV     A,#366Q
02E0 A0          698        MOV     @R0,A          ;R7     CHANNEL A ONLY (PSG#2)
02E1 18          699        INC     R0
02E2 2310        700        MOV     A,#010H
02E4 A0          701        MOV     @R0,A          ;R8
02E5 18          702        INC     R0
02E6 A0          703        MOV     @R0,A          ;R9
02E7 18          704        INC     R0
02E8 A0          705        MOV     @R0,A          ;R10
02E9 18          706        INC     R0
02EA 23A1        707        MOV     A,#161D
02EC A0          708        MOV     @R0,A          ;R11
02ED 18          709        INC     R0
02EE 2303        710        MOV     A,#3D
02F0 A0          711        MOV     @R0,A          ;R12
02F1 18          712        INC     R0
```

```
LOC  OBJ         SEQ        SOURCE STATEMENT

02F2 27          713          CLR     A
02F3 A0          714          MOV     @R0,A          ;P13
                 715
02F4 7440        716          CALL    POPAMP
02F6 7423        717          CALL    LOADP2
                 718
02F8 8A30        719          ORL     P2,#030H
                 720
02FA B82F        721          MOV     R0,#02FH
02FC F0          722          MOV     A,@R0
02FD 07          723          DEC     A
02FE A0          724          MOV     @R0,A
                 725
02FF 93          726          RETR
                 727
                 728
                 729 ;*****************************************************************
                 730
                 731 ;        SUBROUTINE TO LOAD PSG#1 FROM MEMORY LOCATIONS
                 732 ;        20H TO 2DH CORRESPONDING TO REGISTERS 0 TO 13 OF PSG#1.
                 733
0300 9A7F        734 LOADP1:  ANL     P2,#07FH
                 735
0302 C5          736          SEL     RB0
0303 B920        737          MOV     R1,#020H
0305 BA00        738          MOV     R2,#0
0307 BB0E        739          MOV     R3,#14D
                 740
0309 741A        741 JACKO    CALL    LADP1
030B EB16        742          DJNZ    R3,THERE
030D 8A80        743          ORL     P2,#080H
030F B92D        744          MOV     R1,#02DH
0311 27          745          CLR     A
0312 A1          746          MOV     @R1,A
0313 19          747          INC     R1
0314 A1          748          MOV     @R1,A
                 749
0315 93          750          RETR
                 751
0316 19          752 THERE    INC     R1
0317 1A          753          INC     R2
0318 6409        754          JMP     JACKO
                 755
031A 8920        756 LADP1:   ORL     P1,#020H
031C FA          757          MOV     A,R2
031D 90          758          MOVX    @R0,A
031E 99DF        759          ANL     P1,#0DFH
0320 F1          760          MOV     A,@R1
0321 90          761          MOVX    @R0,A
0322 93          762          RETR
                 763
                 764
                 765 ;*****************************************************************
                 766
                 767 ;        SUBROUTINE TO LOAD PSG#2 FROM MEMORY LOCATIONS
```

```
LOC  OBJ         SEQ         SOURCE STATEMENT

                 768 ;        30 TO 3D CORRESPONDING TO REGISTERS 0 TO 14 OF PSG#2
                 769
0323 99F7        770 LOADP2  ANL     P1,#0F7H
                 771
0325 C5          772         SEL     RB0
0326 B930        773         MOV     R1,#030H
0328 BA00        774         MOV     R2,#0
032A BB0E        775         MOV     R3,#14D
                 776
032C 7437        777 LAPOR:  CALL    LADP2
032E EB33        778         DJNZ    R3,HERE
0330 8908        779         ORL     P1,#08H
                 780
0332 93          781         RETR
                 782
0333 19          783 HERE:   INC     R1
0334 1A          784         INC     R2
0335 642C        785         JMP     LAPOR
                 786
0337 8940        787 LADP2:  ORL     P1,#040H
0339 FA          788         MOV     A,R2
033A 90          789         MOVX    @R0,A
033B 99BF        790         ANL     P1,#0BFH
033D F1          791         MOV     A,@R1
033E 90          792         MOVX    @R0,A
033F 93          793         RETR
                 794
                 795
                 796 ;***********************************************************************
                 797
                 798 ;        DECODE REGISTER R6 TO DETERMINE PROPER AMPLITUDE FOR A
                 799 ;        THRUSTER ROCKET SOUND, 1 OF 13 POSSIBLE AMPLITUDES DECAYING
                 800 ;        LOGARITHMICALLY
                 801
0340 C5          802 ROPAMP  SEL     RB0
0341 FE          803         MOV     A,R6
0342 530F        804         ANL     A,#0FH
0344 7268        805         JB3     ROPA
0346 5252        806         JB2     ROPB
0348 124E        807         JB0     ROPC
034A 23B8        808         MOV     A,#0B8H          ; R6=2, LEVEL=3
034C 648E        809         JMP     CAMPO
034E 23B1        810 ROPC    MOV     A,#0B1H          ;   =3,       =4
0350 648E        811         JMP     CAMPO
                 812
0352 325E        813 ROPB    JB1     ROPD
0354 125A        814         JB0     ROPE
0356 23A4        815         MOV     A,#0A4H          ; R6=4, LEVEL=5
0358 648E        816         JMP     CAMPO
035A 23A2        817 ROPE    MOV     A,#0A2H          ;   =5,       =6
035C 648E        818         JMP     CAMPO
035E 1264        819 ROPD    JB0     ROPF
0360 23A1        820         MOV     A,#0A1H          ;   =6,       =7
0362 648E        821         JMP     CAMPO
0364 2398        822 ROPF    MOV     A,#098H          ;   =7,       =8
```

```
 LOC  OBJ         SEQ        SOURCE STATEMENT

 0366 648E        823            JMP     CAMPO
                  824
 0368 5280        825 ROPA       JB2     ROPG
 036A 3276        826            JB1     ROPH
 036C 127C        827            JB0     ROPI
 036E 2394        828            MOV     A,#094H          ; R6=8, LEVEL=9
 0370 648E        829            JMP     CAMPO
 0372 2392        830 ROPI:      MOV     A,#092H          ;    =9,       =10
 0374 648E        831            JMP     CAMPO
 0376 127C        832 ROPH:      JB0     ROPJ
 0378 2391        833            MOV     A,#091H          ;    =10,      =11
 037A 648E        834            JMP     CAMPO
 037C 2388        835 ROPJ:      MOV     A,#088H          ;    =11,      =12
 037E 648E        836            JMP     CAMPO
                  837
 0380 328C        838 ROPG       JB1     ROPK
 0382 1288        839            JB0     ROPL
 0384 2384        840            MOV     A,#084H          ; R6=12, LEVEL=13
 0386 648E        841            JMP     CAMPO
 0388 2382        842 ROPL:      MOV     A,#082H          ;    =13,      =14
 038A 648E        843            JMP     CAMPO
 038C 2381        844 ROPK:      MOV     A,#081H          ;    =14,15,   =15
                  845
 038E 3D          846 CAMPO      MOVD    P5,A
 038F 47          847            SWAP    A
 0390 3C          848            MOVD    P4,A
 0391 93          849 CHIK:      RETR
                  850
                  851
                  852 ;************************************************************************
                  953
                  954 ;      THIS ROUTINE DECODES REGISTER R6 TO DETERMINE IF THE MISSILE
                  855 ;      IS IN THE LOW RANGE (LESS THAN 333 METERS), MID RANGE (333 TO
                  856 ;      666 METERS), OR HIGH RANGE (GREATER THAN 666 METERS).
                  857
                  858 ;      THE AMPLITUDES OF 3 SUCCESSIVE EXPLOSIONS ARE SET FOR A TARGET
                  859 ;      HIT.  A MISSED TARGET PRODUCES ONLY THE FIRST TWO EXPLOSIONS.
                  860
 0392 C5          861 HITAMP     SEL     RB0
 0393 FE          862            MOV     A,R6
 0394 72AE        863            JB3     HADU
 0396 52A1        864            JB2     HITU
                  865
 0398 FD          866 MARY       MOV     A,R5
 0399 07          867            DEC     A
 039A C6[illegible] 968          JZ      JOHN3            ; R6=2,3,4,5   3RD EXPLO    HIGH RANGE
 039C 07          869            DEC     A
 039D C6C5        870            JZ      JOHN2            ;              2ND EXPLO
 039F 64C1        871            JMP     JOHN1            ;              1ST EXPLO
                  872
 03A1 32A5        873 HITU       JB1     HOAX
 03A3 6498        874            JMP     MARY
                  875
 03A5 FD          876 HOAX       MOV     A,R5
 03A6 07          877            DEC     A
```

```
LOC  OBJ        SEQ       SOURCE STATEMENT

03A7 C6CD        878          JZ      JOHN4          ; R6=6, . ,10   3RD EXPLO    MID RANGE
03A9 07          879          DEC     A
03AA C6C5        880          JZ      JOHN2          ;               2ND EXPLO
03AC 64C1        881          JMP     JOHN1                          1ST EXPLO
                 882
03AE 52B8        883 HADU     JB2     HOPE
03B0 32B4        884          JB1     HADE
03B2 64A5        885          JMP     HOAX
03B4 12B8        886 HADE:    JB0     HOPE
03B6 64A5        887          JMP     HOAX
                 888
03B8 FD          889 HOPE     MOV     A,R5
03B9 07          890          DEC     A
03BA C6CD        891          JZ      JOHN4          ; R6=11, . ,15   3RD EXPLO    LOW RANGE
03BC 07          892          DEC     A
03BD C6C9        893          JZ      JOHN3          ;                2ND EXPLO
03BF 64C5        894          JMP     JOHN2          ;                1ST EXPLO
                 895
03C1 230F        896 JOHN1    MOV     A,#0FH         ; SOUND LEVEL 1 (LOWEST)
03C3 3C          897          MOVD    P4,A
03C4 93          898          RETR
03C5 230E        899 JOHN2    MOV     A,#0EH         ; SOUND LEVEL 2
03C7 3C          900          MOVD    P4,A
03C8 93          901          RETR
03C9 230D        902 JOHN3    MOV     A,#0DH         ; SOUND LEVEL 3
03CB 3C          903          MOVD    P4,A
03CC 93          904          RETR
03CD 230C        905 JOHN4    MOV     A,#0CH         ; SOUND LEVEL 4 (LOUDEST)
03CF 3C          906          MOVD    P4,A
03D0 93          907          RETR
                 908
                 909
                 910
                 911
                 912
                 913
                 914
                 915          END

USER SYMBOLS
AOFIN  0072   ANIT   0009   BAT0   0215   BAT01  01D9   BAT1   020F   BAT11  01D7   BAT2   0207   BAT21  01D5
BAT3   01FF   BAT4   01F7   BAT5   01EF   BAT6   01E6   BAT7   01DF   BEA    021E   BEB    0225   BIT0   01AD
BIT1   01A5   BIT2   019C   BIT3   0191   BIT4   0186   BIT5   017B   BIT6   016C   BIT7   0160   BRAVE  0271
CAMAF  026D   CAKET  027E   CAMPO  038E   CHAK   00A0   CHECK  0094   CHIK   0391   COKOP  0275   DALAY  00BF
DECODE 00D1   DELAY  00B4   DEXAA  00A7   DEXO   00A1   DLY    00C1   DRAGO  00F3   DRAGON 0228   DRANG  003D
DUDF   00F8   DUDEX  0277   DUDEX1 00EB   DUDEXO 0100   FLIP   001E   GETTA  00AF   HADE   03B4   HADU   03AE
HERE   0333   HEXO   00AB   HITAMP 0392   HITE   0103   HITE1  00EE   HITEX  029A   HITU   03A1   HORX   03A5
HOPE   03B8   INFIN  0046   INIT   000B   JACKO  0309   JIERT  029E   JOHN1  03C1   JOHN2  03C5   JOHN3  03C9
JOHN4  03CD   JP0    015C   JP1    015A   JP2    0158   JP3    0156   JP4    0154   JP5    0152   JP6    0150
LAHUT  029E   LADP1  031A   LADP2  0337   LAPOR  032C   LOADP1 0300   LOADP2 0323   LODRAM 023B   LOOP1  0086
LOOP2  00E8   MARY   0398   OBLO   00CE   PASS   0115   PB0    01D1   PB1    01CF   PB2    01CD   PB3    01CB
PB4    01C9   PB5    01C7   PB6    01C5   PBY    01DB   POKER  011E   POSS   0112   PUPP   0117   PUPPET 011A
PUSS   010D   RESPG  001A   ROPA   0368   ROPAMP 0340   ROPB   0352   ROPC   034E   ROPD   035E   ROPE   035A
ROPF   0364   ROPG   0380   ROPH   0376   ROPI   0372   ROPJ   037C   ROPK   038C   ROPL   0388   ROPO   0105
ROPOD  00F1   ROPOP  0288   SAFON  007E   SAVOB  0087   SINFIN 0076   TAP1   021A   TAP    0142   TAPAP  0182
```

```
TAPE   014A    TARC   0137    TESTR4 014C    THERE  0316    TIMER  00C6    TRAMP  0217    TRAP   01C0    TRIP   01BE
  TR45 0128    UNFIN  0063

ASSEMBLY COMPLETE,  NO ERRORS
```

```
ASM48 TURRET SRC

ISIS-II MCS-48/UPI-41 MACRO ASSEMBLER, V2.0           PAGE    1

 LOC  OBJ          SEQ         SOURCE STATEMENT

                    1
                    2
                    3 ;            16 SEPTEMBER 1981
                    4
                    5
                    6 ;      THIS IS A PROGRAM TO TURN THE TURRET ON A TANK MODEL
                    7 ;      90 DEGREES, HOLD IT THERE FOR 1.5 SECONDS, AND RETURN IT
                    8
                    9
                   10        INITIALIZATION
                   11
 0000              12        ORG     0
 0000 0409         13        JMP     INIT
 0003              14        ORG     3
 0003 0419         15        JMP     TURN
 0009              16        ORG     9
 0009 [illegible]  17 INIT   MOV     A,#0FFH
 000B [illegible]  18        OUTL    P1,A
 000C 99F7         19        ANL     P1,#0F7H                ;RESET PSG
 000E 8908         20        ORL     P1,#8
 0010 [illegible]  21        ENT0    CLK
 0011 05           22        EN      I
 0012 09           23 TRIG:  IN      A,P1
 0013 9212         24        JB4     TRIG                    ;TEST P14 FOR TRIGGER PULSE
 0015 14EB         25        CALL    GYRO
 0017 0412         26        JMP     TRIG
                   27
                   28
                   29 ;      RUN TURRET FORWARD
                   30
 0019 99FD         31 TURN   ANL     P1,#0FDH                ; 1111/1101 = PORT 1
 001B BAC8         32        MOV     R2,#200D
 001D 14[illegible] 33       CALL    DELAY
 001F 561F         34 WHITE1 JT1     WHITE1
 0021 8906         35        ORL     P1,#06
                   36
 [illegible] 144[illegible] 37  CALL  SOUND
                   38
 [illegible]       39        MOV     R2,#100D
 [illegible]       40        CALL    DELAY
                   41
                   42
                   43        RUN TURRET REVERSE
                   44
                   45        ANL     P1,#0FBH                ; 1111/1011 = PORT 1
                             MOV     R2,#200D
                             [illegible]
                             [illegible]
                             [illegible]
```

```
LOC  OBJ         SEQ        SOURCE STATEMENT

0034 B908         53 DELAY:  MOV     R1,#8                     ; DELAY = R2 X .01 SEC
0036 B8FF         54 LOOP1   MOV     R0,#0FFH
0038 E838         55 LOOP2:  DJNZ    R0,LOOP2
003A E936         56         DJNZ    R1,LOOP1
003C EA34         57         DJNZ    R2,DELAY
003E 83           58         RET
                  59
003F B8C8         60 DALAY:  MOV     R0,#0C8H                  ; DELAY = R2 X .001 SEC
0041 E841         61 DAY     DJNZ    R0,DAY
0043 EA3F         62         DJNZ    R2,DALAY
0045 83           63         RET
                  64
                  65
                  66 ;       THIS IS THE SOUND MAKING ROUTINE
                  67
0046 99F7         68 SOUND   ANL     P1,#0F7H                  ; ALL PSG REGISTEPS GET 0
0048 8908         69         ORL     P1,#8
                  70
004A BA32         71         MOV     R2,#50D
004C 1434         72         CALL    DELAY
                  73
004E B830         74         MOV     R0,#030H
0050 2300         75         MOV     A,#0
0052 A0           76         MOV     @R0,A                     ; R0
0053 18           77         INC     R0
0054 2336         78         MOV     A,#066D
0056 A0           79         MOV     @R0,A                     ; R7
0057 18           80         INC     R0
0058 2307         81         MOV     A,#07H
005A A0           82         MOV     @R0,A                     ; R10
                  83
005B BF48         84         MOV     R7,#048H                  ; LOOP COUNTER
005D B830         85         MOV     R0,#030H
                  86
005F 8901         87 WHISTL  ORL     P1,#1
0061 27           88         CLR     A
0062 90           89         MOVX    @R0,A
0063 99FE         90         ANL     P1,#0FEH
0065 F0           91         MOV     A,@R0
0066 90           92         MOVX    @R0,A                     ; WRITE TO REGISTER R6
                  93
0067 8901         94         ORL     P1,#1
0069 2301         95         MOV     A,#1
006B 90           96         MOVX    @R0,A
006C 99FE         97         ANL     P1,#0FEH
006E 2301         98         MOV     A,#1
0070 90           99         MOVX    @R0,A                     ;                   R1
                 100
0071 8901        101         ORL     P1,#1
0073 2306        102         MOV     A,#6
0075 90          103         MOVX    @R0,A
0076 99FF        104         ANL     P1,#0FEH
0078 230F        105         MOV     A,#170
007A 90          106         MOVX    @R0,A                     ;                   R6
                 107
```

D. B

```
LOC  OBJ          SEQ        SOURCE STATEMENT

007B 8901          108          ORL     P1,#1
007D 2307          109          MOV     A,#7
007F 90            110          MOVX    @R0,A
0080 99FE          111          ANL     P1,#0FEH
0082 18            112          INC     R0
0083 F0            113          MOV     A,@R0
0084 90            114          MOVX    @R0,A                          ; R7
                   115
0085 8901          116          ORL     P1,#1
0087 2308          117          MOV     A,#100
0089 90            118          MOVX    @R0,A
008A 99FE          119          ANL     P1,#0FEH
008C 18            120          INC     R0
008D F0            121          MOV     A,@R0
008E 90            122          MOVX    @R0,A                          ; R10
                   123
008F BA07          124          MOV     R2,#7
0091 143F          125          CALL    DALAY            ; 7 MSEC PER STEP
                   126
0093 EF97          127          DJNZ    R7,CONT
0095 049C          128          JMP     EXPLO
0097 B830          129 CONT     MOV     R0,#030H
0099 10            130          INC     @R0
009A 045F          131          JMP     WHISTL
                   132
009C 8901          133 EXPLO    ORL     P1,#1
009E 27            134          CLR     A
009F 90            135          MOVX    @R0,A
00A0 99FE          136          ANL     P1,#0FEH
00A2 2350          137          MOV     A,#1350
00A4 90            138          MOVX    @R0,A            ; WRITE TO REGISTER R0
                   139
00A5 8901          140          ORL     P1,#1
00A7 2301          141          MOV     A,#1
00A9 90            142          MOVX    @R0,A
00AA 99FE          143          ANL     P1,#0FEH
00AC 2300          144          MOV     A,#150
00AE 90            145          MOVX    @R0,A                          ; R1
                   146
00AF 8901          147          ORL     P1,#1
00B1 2306          148          MOV     A,#6
00B3 90            149          MOVX    @R0,A
00B4 99FE          150          ANL     P1,#0FEH
00B6 231F          151          MOV     A,#370
00B8 90            152          MOVX    @R0,A                          ; R6
                   153
00B9 8901          154          ORL     P1,#1
00BB 2307          155          MOV     A,#7
00BD 90            156          MOVX    @R0,A
00BE 99FE          157          ANL     P1,#0FEH
00C0 2316          158          MOV     A,#0560
00C2 90            159          MOVX    @R0,A                          ; R7
                   160
00C3 8901          161          ORL     P1,#1
00C5 2308          162          MOV     A,#8
```

```
LOC  OBJ          SEQ        SOURCE STATEMENT

00C7 90            163        MOVX   @R0,A
00C8 99FE          164        ANL    P1,#0FEH
00CA 2310          165        MOV    A,#020Q
00CC 90            166        MOVX   @R0,A                    ;             R10
                   167
00CD 8901          168        ORL    P1,#1
00CF 230B          169        MOV    A,#13Q
00D1 90            170        MOVX   @R0,A
00D2 99FE          171        ANL    P1,#0FEH
00D4 23FF          172        MOV    A,#0FFH
00D6 90            173        MOVX   @R0,A                    ;             R13
                   174
00D7 8901          175        ORL    P1,#1
00D9 230C          176        MOV    A,#14Q
00DB 90            177        MOVX   @R0,A
00DC 99FE          178        ANL    P1,#0FEH
00DE 233F          179        MOV    A,#077Q
00E0 90            180        MOVX   @R0,A                    ;             R14
                   181
00E1 8901          182        ORL    P1,#1
00E3 230D          183        MOV    A,#15Q
00E5 90            184        MOVX   @R0,A
00E6 99FE          185        ANL    P1,#0FEH
00E8 27            186        CLR    A
00E9 90            187        MOVX   @R0,A                    ;             R15
                   188
00EA 83            189        RET
                   190
                   191
00EB BA0A          192 GYRO   MOV    P2,#10D
00ED 1434          193        CALL   DELAY
                   194
00EF 8901          195        ORL    P1,#1
00F1 2302          196        MOV    A,#2
00F3 90            197        MOVX   @R0,A
00F4 99FE          198        ANL    P1,#0FEH
00F6 23FF          199        MOV    A,#0FFH
00F8 90            200        MOVX   @R0,A                    ;             R2
                   201
00F9 8901          202        ORL    P1,#1
00FB 2303          203        MOV    A,#3
00FD 90            204        MOVX   @R0,A
00FE 99FE          205        ANL    P1,#0FEH
0100 2303          206        MOV    A,#3
0102 90            207        MOVX   @R0,A                    ; B = 4XFREQ  R3
                   208
0103 8901          209        ORL    P1,#1
0105 2304          210        MOV    A,#4
0107 90            211        MOVX   @R0,A
0108 99FE          212        ANL    P1,#0FEH
010A 23FF          213        MOV    A,#0FFH
010C 90            214        MOVX   @R0,A                    ;             R4
                   215
010D 8901          216        ORL    P1,#1
010F 2305          217        MOV    A,#5
```

```
 LOC  OBJ         SEQ        SOURCE STATEMENT

 0111 90           218         MOVX     @R0,A
 0112 99FE         219         ANL      P1,#0FEH
 0114 230F         220         MOV      A,#0FH
 0116 90           221         MOVX     @R0,A                    ; C = 1XFREQ       R5
                   222
 0117 8901         223         ORL      P1,#1
 0119 2306         224         MOV      A,#6
 011B 90           225         MOVX     @R0,A
 011C 99FE         226         ANL      P1,#0FEH
 011E 230F         227         MOV      A,#0FH
 0120 90           228         MOVX     @R0,A                    ;                  R6
                   229
 0121 8901         230         ORL      P1,#1
 0123 2307         231         MOV      A,#7
 0125 90           232         MOVX     @R0,A
 0126 99FE         233         ANL      P1,#0FEH
 0128 2309         234         MOV      A,#3110
 012A 90           235         MOVX     @R0,A                    ;                  R7
                   236
 012B 8901         237         ORL      P1,#1
 012D 2309         238         MOV      A,#110
 012F 90           239         MOVX     @R0,A
 0130 99FE         240         ANL      P1,#0FEH
 0132 230F         241         MOV      A,#0FH
 0134 90           242         MOVX     @R0,A                    ;                  R11
                   243
 0135 8901         244         ORL      P1,#1
 0137 230A         245         MOV      A,#120
 0139 90           246         MOVX     @R0,A
 013A 99FE         247         ANL      P1,#0FEH
 013C 230F         248         MOV      A,#0FH
 013E 90           249         MOVX     @R0,A                    ;                  R12
                   250
 013F BA32         251         MOV      R2,#500
 0141 1434         252         CALL     DELAY
                   253
 0143 99F7         254         ANL      P1,#0F7H
 0145 8908         255         ORL      P1,#8
                   256
 0147 83           257         RET
                   258
                   259
                   260
                   261
                   262
                   263         END

USER SYMBOLS
CONT   0097   DALAY  003F   DAY    0041   DELAY  0034   EXPLO  009C   GYRO   00EB   INIT   0009   LOOP1  0036
LOOP2  0038   SOUND  0046   TRIG   0012   TURN   0019   WHISTL 005F   WHITE1 001F   WHITE2 002F

ASSEMBLY COMPLETE,  NO ERRORS
```